2017

A Lebesgue Sampling based Diagnosis and Prognosis Methodology with Application to Lithium-ion Batteries

Wuzhao Yan
University of South Carolina

A Lebesgue Sampling based Diagnosis and Prognosis Methodology with Application to Lithium-ion Batteries

by

Wuzhao Yan

Bachelor of Science
University of Science and Technology of China 2005
Doctor of Philosophy
University of Science and Technology of China 2010

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Electrical Engineering

College of Engineering and Computing

University of South Carolina

2017

Accepted by:

Bin Zhang, Major Professor

Xiaofeng Wang, Committee Member

Hebert Ginn, Committee Member

Paul Ziehl, Committee Member

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

Dedication

This dissertation dedicates to my parents and my wife, for their love and support all the time. This dissertation also dedicates to grandma, who passed away during the pursuing of my Ph.D. degree.

Acknowledgments

Four-years Ph.D. study experience in University of South Carolina is a hard but worthy journey in my life. This dissertation could not have been finished without the support from many people. It is my great pleasure to acknowledge those who have given me guidance, assistance, and companionship.

Foremost, I would like to express my most sincere gratitude to my advisor, Dr. Bin Zhang, for giving me an opportunity to participate in his research group. It has been an honor to be his Ph.D. student. His inspiring ideas, broad scientific knowledge and illuminating instructions always lead me in all the time of my research work. His enthusiastic encouragement and patient guidance on both research and my career have been invaluable. He is not only a advisor but also a good friend in my life, I could not have imagined having a better advisor and friendlier mentor for my Ph.D study.

Second, I would also like to acknowledge my committee members Dr. Xiaofeng Wang, Dr. Herbert Ginn, and Dr. Paul Ziehl, for their time and efforts in service on my Ph.D. committee and for their brilliant comments and valuable suggestions in the proposal defense and dissertation.

Thanks to all my group members: Zhichao Liu who helped me a lot in control system model design, John Weddington who helped me in fusion theory; Guangxing Niu who helped me in understanding neural network method; and Shijie Tang who helps me in preparing slides.

Thanks to all my office mates: Zheqing Zhou, Tengxing Wang, Wei Jiang, Lixing Yang, Dr. Yujia Peng, Dr. David Coats, Dr. Moinul Islam, Dr. Amin Ghaderi,

and Tim brüdigam for the friendly working environment and for having my research experience rich and colorful with the valuable time working and having fun together. Many thanks to friends who helped me integrate into the oversea school life and make it memorable.

Thanks to all professors who had taught me and helped me: Dr. Bin Zhang, Dr. Xiaofeng Wang, Dr. Guoan Wang, Dr. Asif Khan, Dr. Gabriel Terejanu, Dr. Enrico Santi. Thanks to all the department staffs for their wonderful assistant in providing a friendly and nice environment. Also thanks a lot to my friends in Department of Computer Science: Dazhou Guo helped me in using LaTEX, Shizhou Han taught me coding skills in Matlab, Zhonghao Liu taught me coding skill in Java, Haozhou Yu helped me in computer knowledge.

Thanks to my friends, who used to have lunch break with me. Jiting Xu, Can Chen, Haixia Li, Yue He, Ruofan Xia, Dr. Dan Li, Dr. Zhenhua Tian. We had a lot of fun together, thank you all.

Thanks to my friends: Dr. Michael Murphy, Katrina Li, Chunling Wang, Lingmei Zhang, Danfeng Liu, Xinbin Yu, Tie Yan, Xiaonan Ma, Qing Lei, and so on. We had beautiful time together. Thanks to Dr. Cong Ren, Dr. Peiyi Wang, Dr. Wei Zhao, Jingyuan Chen, who take care of me when I was injured. Thanks to my friends at USC and Chinese Soccer Team, appreciate the time we spent together in our golden age: Dr. Meng Guo, Dr. Sheng Feng, Dr. Siwei Wang, Dr. Xiaoguang Ma, Dr. Chao Li, Dr. Kejian Yao, Dr.Wei Zhao, Bo Zhao, Yong Li, Wanze Xie, Yuleng Zeng, Dr. Kangmin Xie, Huidong Zhou, Dr. Yongqiang Chu, Dr. Diansheng Guo, Dr. Xiaofeng Wang, Xin Liu, Zhongwen Luo, Qun Fang, Yichao Lin, Yan Zhang, Dr. Yu Chen, Dr. Longfei Ye, Jie Cai, Zhihao Zhang, Yuan Xu, Yicheng Zhu, Dr. Chao Jiang, Songyuan Deng, and so on.

Thanks to my friends at USC badminton Team, I really enjoy the games we had. Dr. Xinyu Huang, Dr. Xinfeng Liu, Dr. Richard Cheng, Dr. Shou Liu, Dr. Yuchen

Mao, Yuan Shen, Tiejun Zhang, Guanghan Huang, Yanan Zhang, Jian Wu, Zhichao Peng, Xiyu Lan, Lixin Che, and so on.

Finally, I would like to express my eternal gratitude to my parents, my wife, my grandparents, younger sisters, and all other family members for their everlasting love and unlimited support. It is their constant encouragement and love that help me go through the hard times of study and enjoy the life with a carefree mind.

Abstract

Fault diagnosis and prognosis (FDP) plays an important role in the modern complex industrial systems to maintain their reliability, safety, and availability. Diagnosis aims to monitor the fault state of the component or the system in real-time. Prognosis refers to the generation of long-term predictions that describe the evolution of a fault and the estimation of the remaining useful life (RUL) of a failing component or subsystem.

Traditional Riemann sampling-based FDP (RS-FDP) takes samples and executes algorithms in periodic time intervals and, in most cases, requires significant computational resources. This makes it difficult or even impossible to implement RS-FDP algorithms on hardware with very limited computational capabilities, such as embedded systems that are widely used in industries.

To overcome this bottleneck, this proposal develops a novel Lebesgue sampling-based FDP (LS-FDP), in which FDP algorithms are implemented "as-neede". Different from RS-FDP, LS-FDP divides the state axis by a number of predefined states (also called Lebesgue states). The computation of LS-based diagnosis is triggered only when the value of measurements changes from one Lebesgue state to another, or "event-triggered". This method significantly reduces the computation demands by eliminating unnecessary computation. This LS-FDP design is generic and able to accommodate different algorithms, such as Kalman filter and its variations, particle filter, relevant vector machine, etc.

This proposal first develops a particle filtering based LS-FDP for li-ion battery applications. To improve the accuracy and precision of the diagnosis and prognosis

results, the parameters in the models are treated as time-varying ones and adjusted online by a recursive least square (RLS) method to accommodate the changing of dynamics, operation condition, and environment in the real cases. Uncertainty management is studied in LS-FDP to handle the uncertainties from inaccurate model structure and parameter, measurement noise, process noise, and unknown future loading.

The extended Kalman filter implemented in the framework of LS-FDP yields a more efficient LS-EKF algorithm. The proposed method takes full advantage of EKF and Lebesgue sampling to alleviate computation requirements and make it possible to be deployed on most of the distributed FDP systems.

All the proposed methods are verified by a study with the estimation of the state of health and RUL prediction of Lithium-ion batteries. The comparisons between traditional RS-FDP methods and LS-FDP show that LS-FDP has a much lower requirement on the computational resource. The proposed parameter adaptation and uncertainty management methods can produce more accurate and precise diagnostic and prognostic results. This research opens a new chapter for FDP method and make it easier to deploy FDP algorithms on the complicate systems build by embedded subsystem and micro-controllers with limited computational resources and communication band width.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Modern engineering systems, such as aircrafts, industrial plants, power stations, and electrical vehicles are designed to perform desired functions with requirements of reliability, productivity, safety, and availability. However, the system will deteriorate and the performance will degrade with the increase of storage time and service time. To keep the performance of the system, fault diagnosis and prognosis (FDP) is developed to estimate the fault state, predict the remaining useful life (RUL) [1–5], and provide appropriate information for maintenance decision.

FDP can be divided into two catergories: centralized FDP and distributed FDP. Centralized FDP is implemented on a central computer. Data collected from distributed sensors are sent to the central computer for data-processing, feature extraction, and FDP algorithm execution. This requires high communication, power, and computation capabilities. The rapid increase of system complexity put more challenges on the centralized FDP design and the distributed FDP gets widely accepted in engineering design [6–10]. With this trend, more and more FDP functions are deployed on local processors, which are often microprocesor and embedded systems, to alleviate the requirements on communication bandwidth, power, and computation, thus to improve the reliability of the whole system [11–14].

Traditional Riemann sampling-based FDP (RS-FDP) takes samples and executes algorithms in periodic time intervals [15–17] and, in most cases, requires significant

computational resources. This makes it difficult or even impossible to implement RS-FDP algorithms on these local microprocessors and embedded systems with very limited computational capabilities. This becomes the bottleneck that prevents the distribution of FDP algorithms in complex systems and cost-efficient FDP solutions must be developed.

To overcome this bottleneck, we propose a Lebesgue sampling-based FDP (LS-FDP) framework, which employs an efficient FDP philosophy where computation can be executed on an "as-needed" basis. The novelty comes from the concept of "Lebesgue sampling (LS)" (or "event-based sampling"), which divides the state axis by a number of predefined states (also called Lebesgue states). In contrast to conventional approaches in which FDP is executed periodically, LS-FDP is triggered only when an event takes place, which indicates the fault state has changes and justify the implementation of FDP calculation. With the feature of "execution only when necessary" in Lebesgue sampling (LS), the computation efforts in LS-FDP can be significantly reduced by eliminating unnecessary computation.

The output of diagnosis is the estimation of fault state given by a probability density function, which is compared against the baseline data to achieve the probability of fault detection. When LS-based prognosis is activated, prognostic algorithm is executed based on the Lebesgue sampling model (LSM) to estimate distributions of operating time for the fault state reaching each Lebesgue state. The prediction horizon of LS-FDP is defined on the fault dimension axis and described by the number of Lebesgue states. This eliminates the recursive state estimation in RS-based prognosis and provides a straightforward means to conduct prognosis that requires little computation resources.

1.1.1 Lithium-ion Batteries

Battery is a safety critical component that provides power to system functions including command, control, communications, computers, and intelligence [18, 19]. Lithium-ion batteries are widely used due to the advantages in high energy density, high cycle life, good resistance to memory, and less weights. It is very important to monitor the states of the batteries and ensure batteries safe, reliable, and cost-efficient. Since the state of the batteries, such as state-of-health (SOH), state-of-charge (SOC), and RUL, are not directly observable, FDP is employed to estimate the functional status of batteries. Our research will focus on the diagnosis and prognosis algorithms development with application to lithium-ion batteries.

1.1.1.1 State of Charge (SOC)

SOC is an indicator that represents the available charge stored in the battery compared to the full capacity charge of the battery. An accurate estimation of the SOC is necessary not only for optimal management of the energy stored in the battery but also to protect the battery from being over-discharged or overcharged, which is extremely harmful for battery life and may lead to potentially dangerous situations. Despite the importance, the SOC cannot be measured directly from the battery terminals.

1.1.1.2 State of Health (SOH)

SOH describes the physical condition of a battery. Unlike SOC, there is no clear-cut definition of SOH. A general definition of SOH is that it quantifies the battery's ability to store energy and deliver specified performance compared to a fresh battery [20]. As the battery ages, the SOH degrades in two main areas. First, the degradation of maximum power output of the battery is used to evaluate the SOH, especially in

the application of hybrid electrical vehicles (HEV). The internal resistance increase due to the growth of solid electrolyte interface (SEI), resulting in a loss of maximum power output of the battery. Therefore, battery power loss can be used to indicate the battery SOH [21, 22]. In this case, SOH is defined as a function of high-frequency resistance of the battery in [21], and is described by the estimated resistance.

Second, the battery capacity is used as the indicator of SOH [23–27] in the applications of consumer electronics and electrical vehicles (EV). Considerable loss of battery capacity will lead to ineffective battery operation and reduced service time.

1.1.1.3 Remaining Useful Life (RUL)

RUL is the forecast of battery SOH, which usual is represented by charging-discharging cycles before battery SOH reaches a predefined threshold. Typical RUL estimation is conducted by modeling the degradation of the battery performance to predict battery end-of-life (EOL), which is often defined as the battery capacity reduces to 80% of its original capacity [28].

1.2 Fault Diagnosis and Prognosis

Battery FDP aims to estimate SOC and predict SOH and RUL. By these means, the state of the battery can be monitored to avoid hazardous operations and schedule the missions for the battery-powered systems.

FDP are not new concepts [11, 29–32]. Diagnosis aims to monitor the health state of the component or the system, detect, and isolate impending faults in real-time. Prognosis predicts the evolution of a fault and the RUL of a failing component or subsystem.

For the application of batteries, the capacity starts to degrade from the beginning of its service, the diagnosis is executed to monitor the states of the battery. The

primary mission of failure prognosis is to describe the capacity degradation and estimate the RUL or time-of-failure (ToF) of the batteries.

One of the key concepts within the FDP framework is the state/RUL probability density function (pdf). Figure 1.1 illustrates the key concepts of the pdf and explain the diagnosis process as follows:

1. The baseline (green) of the system can be obtained from measurement of the system when it is in health condition. With requirements from customers, false alarm and confidence level can be defined, such as 5% and 90%, respectively.

2. Based on the baseline pdf and false alarm rate, a threshold of fault detection is determined (blue), beyond which the system is claimed to be faulty.

3. Compare the real-time pdf (red) and the baseline pdf, if 90% of the real-time pdf is beyond the blue threshold, a fault is detected with 5% false alarm rate and 90% confidence.

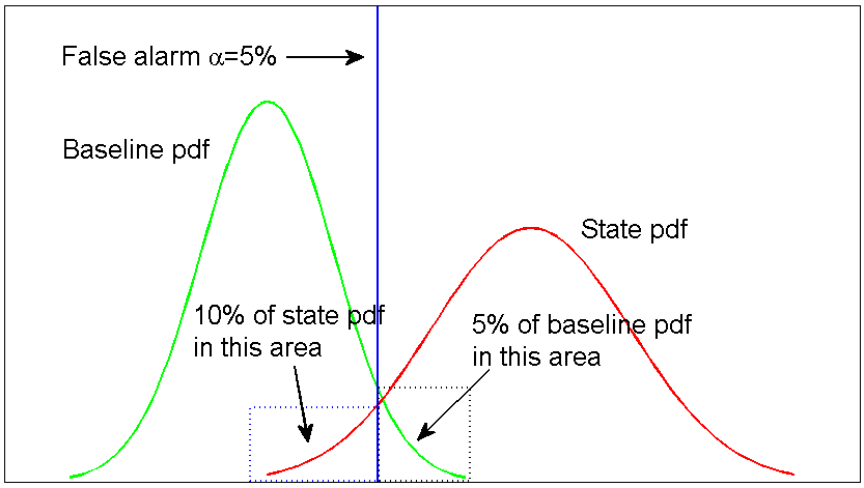


Figure 1.1: Fault detection criteria and fault detection process.

With the state pdf obtained from diagnosis, the RUL pdf is achieved by projecting the state pdf from current time instant t_0 to the future time instant until the predicted state pdf reaches a predefined failure threshold [33]. Figure 1.2 illustrates the process

of RUL pdf prediction. The initial battery capacity at t_0 has a 95% confidence interval of $[c_0,\ c_1]$. For the RUL pdf shown in this figure, t_J is the just-in-time (JIT) point with probability of failure (PoF) =10%, which defines the maximum acceptable failure risk beyond which system can no longer be operated as the risk of failure is deemed excessive. This enables us to implement the principle of JIT maintenance. The expected value of the RUL pdf is t_k, which means there is a 50% chance that the system will fail by time instant t_k. The time interval between fault detection time and actual system failure time is referred to as the lead-time interval (LTI).

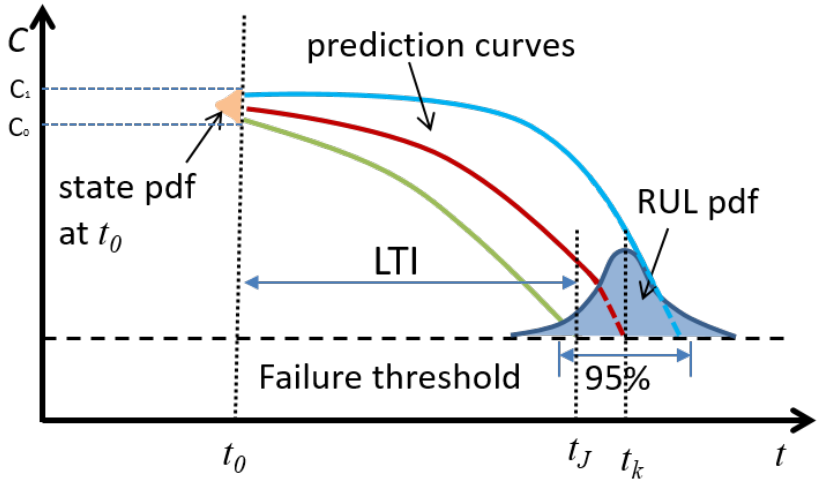


Figure 1.2: Battery capacity extrapolation and the achieved RUL probability density function.

1.3 FDP Approaches

A general accepted method is to divide the FDP approaches into three primary catergories: model-based methods, data-driven methods, and Bayesian filtering methods [34–38]. To make the discussion brief, this proposal will focus on the FDP of Lithium-ion batteries.

1.3.1 Model-based Approaches

Model-based approaches utilize a mathematical tool to model the mechanism of a battery fault in the form of differential equations or equivalent transformations. Different model-based FDP approaches have been developed [39–42].

1.3.1.1 Empirical Model–Equivalent Circuit Model

Equivalent circuit model (ECM) describes the macroscopic behaviour of lithium-ion battery in terms of electrical characteristics. In this model, a lithium-ion battery is considered as a power source with current, voltage, and internal impedance.

The simplest ECM is an ideal linear model [43] consisting of the R and open circuit voltage as a voltage source V_{oc}, as shown in Figure 1.3 (a). This ECM model can only represent the battery working under a constant current load. More complex ECM models are developed to describe the battery more accurate with one or two RC circuits, as shown in Figure 1.3 (b) [44]. This ECM model can illustrate the slow and fast battery charge/discharge response.

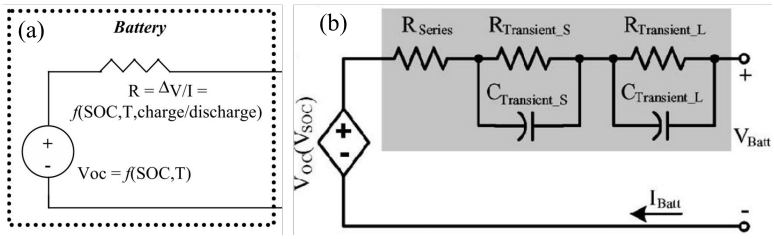


Figure 1.3: Linear resistive battery model (a) without RC and (b) with 2 RC circuit [43, 44]

1.3.1.2 Physical Models

Physical models are established on the basis of mass transfer, chemical thermodynamics and electrodynamics, and many parameters of batteries internal materials are involved, which makes it difficult to obtain SOC/SOH estimation with

accuracy, especially for online applications. Because of the huge computations, these models are usually used for off-line application, such as the battery performance analysis and battery design.

Most of the battery physical models are based on the porous electrode theory. Porous electrode theory is one of the most accurate models to describe the electrochemical reaction inside the battery. However, it is difficult to find an analytical solution for a complete battery model based on porous electrode theory. Appropriate assumptions are introduced to get simplified approximation models, which can be classified into three categories, pseudo 2D (P2D) model [45], single particle model (SP) [46, 47], and porous electrode model (PP) with polynomial approximation [48, 49]. Since these models will not be used in our research, we will not discuss them here. For more details, please refer to the references listed above.

1.3.1.3 Empirical Model

The determination of a complete dynamical model in terms of differential equations may be unnecessary or impractical in some applications. In this situation, empirical models are built based on experiences, which require less detailed information than physical-model-based method and are very effective.

1.3.2 Data-drive Approaches

Data-driven methods are used to train the models above with historical and monitoring data. The data-driven methods include autoregressive moving average (ARMA) model [50–53], fuzzy logic [54], artificial neural network (ANN) [55–57], support vector machine (SVM) [20, 58], relevance vector machine (RVM) [59], etc. Each of these methods have their respeective advantages and disadvantages.

ARMA method is simple and low at computational cost. However, this method lacks of uncertainty representation. Fuzzy logic methods have the ability to describe

the uncertainty in processing the noisy data. The requirement of electrochemical impedance spectrum (EIS) data limits this method to be used for online estimation [54].

Artificial Neural Networks (ANN) requires little physical knowledge of systems and adopts a "black box" approach to handle different sources of data. The limitation of ANN approaches is the absence of ability for uncertainty management, lack to provide probabilistic decision function, requirement of large volume data, incapability to select the best design (number of layers, nodes, and training parameters), and slow-learning and time-consuming with non-uniqueness solution due to convergence to local minima especially for large-order systems [60–62].

Support vector machine (SVM) and relevance vector machine (RVM) are emerging algorithms for battery diagnosis and prognosis. The difference is that RVM is built in Bayesian framework and thus has probabilistic output. The major limitation of SVM is the lack of uncertainty representation [63], whcih is critical in FDP. As for RVM, the major limitation is that its long-term prediction capability is poor such that it is difficult to obtain satisfied RUL prediction by using RVM directly [64, 65].

1.3.3 Bayesian Methods

Bayesian methods employ a fault state dynamic model and a measurement model, along with measurements, to predict the posterior probability density function (pdf) of fault state and predict the time evolution of a fault [66–68]. Various Bayesian estimation techniques have been investigated for diagnosis and prognosis, such as the Kalman Filter (KF), Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF).

Bayesian method is a recursive algorithm that estimates the true state of a system based on noisy measurement. It involves two steps:

1. Prediction step estimates the state value based on the fault dynamic model and

the initial or past fault state.

2. Correction step where the estimated state is adjusted based on the difference between the estimated value and the actual measurement.

KF produces optimal solution for a linear system with Gaussian noise, which is not always true for real-life FDP applications. For nonlinear battery fault or degradation, analytical solution are generally too complicated or even do not exist. To address this problme, EKF is proposed, which linearizes the state model around the mean of the previous state estimate using Taylor expansion. The limitations of EKF are that it does not perform well for highly nonlinear systems, and is sensitive to errors in initial states.

To overcome some of the limitations of EKF, Unscented Kalman Filter (UKF) is proposed [69–72], which uses a set of sample points to ensure accurate estimation of the mean and covariance. EKF and UKF has been used for diagnosis and prognosis of SOC and SOH with different models [73–75].

For highly nonlinear fault dynamics with non-gussian noises, particle filters (PF), which is based on the Bayesian theory and the concept of sequential importance sampling, are often employed [76, 77]. The principle of PF methodology is the approximation of relevant distributions by particles and weights. Compared to classical Monte Carlo methods, PF reduces the number of samples required to approximate the distributions with necessary precision. This principle is of particular benefit in design diagnosis and prognosis algorithms for complex dynamic systems with nonlinear nature and ambiguous operation conditions. PF has been used with ECM model [78, 79], lumped parameter mode [19], Coulomb-counting model [80], various empirical models [81] and data-driven curve fitting models [68] for battery FDP and have shown great performance in various applications.

Performance evaluation plays an importance role in FDP. We will use the following performance indexes to evaluate the proposed approach.

1.4.1 Accuracy and Precision

Accuracy is a measure of how close a state/RUL estimate is to the actual state/RUL. Precision is a measure of the narrowness of an interval in which the state/RUL prediction distribution falls. It is defined on the basis of the variance of the predicted results for many experiments, which can be descried by confidence interval of an estimation distribution. Figure 1.4 (a) illustrates two state distributions that have the same precision but different accuracy. The black one has higher accuracy than the read one since its mean value is close the true value. Figure 1.4 (b) illustrates two state distributions that have the same accuracy but different precision. The red one has higher precision since it has a smaller confidence bounds.

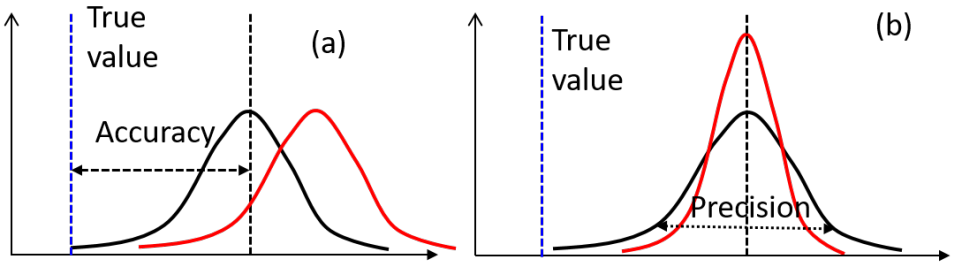


Figure 1.4: Illustration of prediction accuracy and precision [82].

1.4.2 $\alpha - \lambda$ Metric

$\alpha - \lambda$ performance determines whether the prediction falls within specified levels of a performance measure at particular times. Mathematically, this metrics is written as follows and illustrated in Figure 1.5 [83].

$$[1-\alpha]\cdot r^*(t_\lambda) \leq r(t_\lambda) \leq [1+\alpha]\cdot r^*(t_\lambda) \tag{1.1}$$

where α is the accuracy modifier, λ is the window modifier given by $t_\lambda = t_P + \lambda(t_{EOL} - T_P)$, t_P and t_{EOL} are the time instant when prognosis starts and its end of life, $r^*(t_\lambda)$ is the actual RUL, and $r(t_\lambda)$ is the RUL estimation at t_λ.

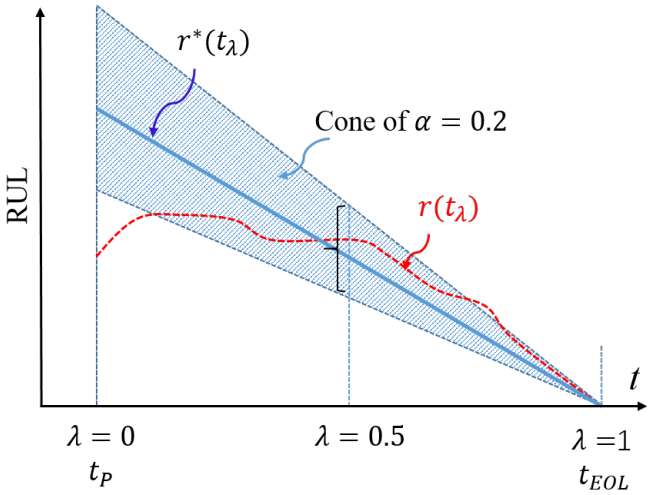


Figure 1.5: $\alpha - \lambda$ accuracy with the accuracy cone shrinking with the time on RUL vs. time plot [83].

Chapter 2

Research Objectives and contributions

2.1 Objective

Distributed FDP design is widely accepted in engineering design, especially for complicated systems [6–10]. With this trend, FDP functions are deployed on local processors, which are often microprocessors and embedded systems [11–14] with very limited computational resources. Since many system functions have been implemented on these local processor, they have even less resources for FDP. On the other hand, traditional RS-based FDP (RS-FDP) takes samples and executes algorithms in periodic time intervals [15–17] and, in most cases, requires significant computational resources. This makes it difficult or even impossible to implement RS-FDP algorithms on these local processors.

To overcome this limitation, LS-FDP is developed. Different from Riemann sampling-based FDP (RS-FDP), LS-FDP divides the state axis by a number of predefined states (also called Lebesgue states). The execution of LS-FDP is triggered only when the value of feature, or condition indicator, changes from one Lebesgue state to another, or an event happens, which significantly reduces the computation demands by eliminating unnecessary computation.

The main contributions of this dissertation are:

1. Develop the concept of Lebesgue sampling into FDP and develops a novel Lebesgue sampling-based FDP approach with an philosophy of "execution only when necessary" or an "as-needed" basis. With this new feature, the

LS-FDP enables the FDP on systems with limited computation capabilities and enables the distributed FDP. The novelties of this work include: 1). Propose a new LS-FDP framework with a complete design methodology, which is able to accommodate different diagnostic and prognostic algorithms; 2) Develop a particle filtering-based FDP algorithm in the LS-FDP framework; and 3) Present an experiment of Lithium-ion battery SOH diagnosis and prognosis with comparison against traditional RS-based approach.

2. Implement the uncertainty management and parameter adaptation in the LS-FDP algorithm to improve the performances of the algorithms. A recursive least square method is used to adjust the parameters of the diagnosis and prognosis models to accommodate the changing of fault dynamics. A feedback loop based on the differences between the predicted time distribution and real-time measurement is used to adjust the noise term in the model. Particle filtering-based algorithms in LS framework with/without parameter adaptation and uncertainty management are developed and verified with an application to the prediction of SOH of Lithium-ion batteries. TTF is more accurate compared with the results achieved without the parameter adaptation. The uncertainty of the predicted TTF is remarkably reduced by the proposed method

3. The Lebesgue state length are kept as constant in our early research, which is not optimal in that it may use more computational resource when fault growth is slow and lead to low performance when fault growth is fast. To address this problem, Lebesgue state length is adaptively adjusted according to the fault growth speed. With adaptive Lebesgue state length, when the fault growth becomes fast, more resources are assigned to FDP so that it will not sacrifice the FDP performance. On the other hand, when the fault growth is slow, the Lebesgue state length is increased to reduce the computation on FDP. With

this method, the FDP is executed in a low frequency when the fault growth is slow, and in a high frequency when the fault growth is fast. This enables the LS-FDP algorithms to monitor the fault state effectively and to optimally distribute the computational resources.

4. Develop EKF in Lebesgue sampling work to reduce the computation cost furthermore, especially for the application of SOC estimation. Different from SOH that describes the capacity degradation in charging-discharging cycles, SOC is more challenging in that it requires real-time estimation and more time sensitive. A LS-EKF based battery management algorithm is developed, which estimates the SOH and update the initial capacity for the SOC estimation. An experiment of Lithium-ion battery SOH diagnosis and prognosis with comparison against traditional RS-EKF approach is presented. During the SOC estimation, the OCV-SOC curve is obtained by using second order ECM model with parameter adaptation since the ECM model parameters changes with the degradation of Lithium-ion battery. It is demonstrated that the proposed approach is able to reduce the requirement on computational sources compared with traditional RS-EKF, and results in accurate SOH and SOC estimation. This proposed approach combines the advantages of EKF and LS method, which results in low computation and small uncertainty accumulation. Moreover, as the SOC data shows different nonlinear dynamics in a large amount charge-discharge cycle, this experiments provide a solid verification and validation (VeVa) of the proposed approach.

2.2 Dissertation Organization

The dissertation is organized as follows: Chapter 3 discusses the basic idea of Lebesgue Sampling (LS) method and implementation of LS-based FDP algorithm (LS-FDP) via PF. The uncertainty management and model parameter adaptation

of the LS-FDP are studied in Chapter 4. Chapter 5 discusses the optimization of Lebesgue length to reduce the computation of FDP when the fault growth is slow and increase the execution of FDP when the fault growth is fast. Chapter 6 develops a complete battery health management system that integrates SOH and SOC in a LS-EKF approach, which not only provide a comprehensive analysis of battery health state, but also verify and validate the proposed approach. Chapter 7 gives the conclusions and future works of this dissertation .

Chapter 3

Lebesgue Sampling-based diagnosis and prognosis

This chapter introduces the concept of Lebesgue sampling in FDP and proposes a Lebesgue sampling based fault diagnosis and prognosis (LS-FDP) framework. In the proposed LS-FDP, a particle filtering approach is developed and presented with experiential results to verify the proposed approach.

3.1 The Proposed LS-FDP Framework

The unique innovative feature of the proposed LS-FDP is that the diagnosis is carried out only when new measurements justify that the fault conditions have changes to warrant the execution. The LS-FDP framework is illustrated in Figure 3.1, which integrates external inputs, Lebesgue samples of feature and fault dimension, event checker, models for diagnosis and prognosis, and diagnostic and prognostic algorithms.

In this Chapter, our focus is the introduction of Lebesgue sampling into diagnosis and prognosis. Therefore, we will not discuss data collection, preprocessing, and feature extraction. After a feature has been successfully extracted from data to indicate the growth of a fault, the performance and efficiency of FDP relies greatly on the dynamic model that describes the fault behavior, and the diagnostic and prognostic algorithms, which will be elaborated in the following sections.

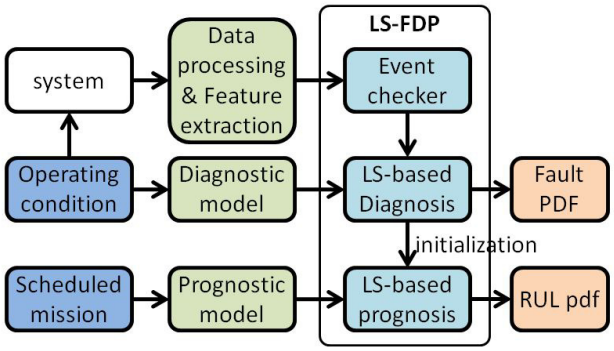


Figure 3.1: The implementation framework of LS-FDP

3.1.1 Fault Mechanism Modeling

Assume that the actual fault growth dynamics can be described by the following continuous-time differential equation:

$$\dot{a} = F(a, u, \omega_a) \tag{3.1}$$

where a is the fault dimension, u is system input including items (such as external environmental factors and operating modes) that have impacts on fault growth, ω_a is the process noise representing modeling uncertainties, and $F(\cdot)$ is a nonlinear function that describes the fault growth under the current fault dimension with input u. The feature or condition indicator, denoted by y, is extracted from raw measurements and serves as the real-time measurement for FDP algorithm. Note that the mapping between y and a can be described by a nonlinear function $y = h(a)$. In most cases, a is not measurable and $y = a$ is employed such that we can use y to indicate fault a directly. To simplify the description, we take $y = a$ in the following discussion.

In general, the discretized realtime model of this continuous-time system can be described by the following equation:

$$\hat{a}(t_{k+1}) = \hat{F}(\hat{a}(t_k), u(t_k)) + \omega_a(t_k) \tag{3.2}$$

where $\hat{F}$ describes the iteration in the state. The Lebesgue state model is described

as:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + D\left(\hat{a}(t_k)\right) \cdot \frac{F\left(\hat{a}(t_k), u(t_k)\right)}{\| F\left(\hat{a}(t_k), u(t_k)\right) \|} \tag{3.3}$$

where $D\left(\hat{a}(t_k)\right)$ is the quantization size that may depend on the state. If it is chosen as a constant, it can be denoted by D directly. Note that $F\left(\hat{a}(t_k), u(t_k)\right) / \| F\left(\hat{a}(t_k), u(t_k)\right) \|$ is a nonlinear term in the model, which is actually the sign of $F\left(\hat{a}(t_k), u(t_k)\right)$ and can be written as $sgn(F\left(\hat{a}(t_k), u(t_k)\right))$. By using Euler method, it can be approximated by $sgn(\hat{F}(\hat{a}(t_k), u(t_k)) - \hat{F}(\hat{a}(t_{k-1}), u(t_{k-1})))$. The model derivation is shown in Figure 3.2.

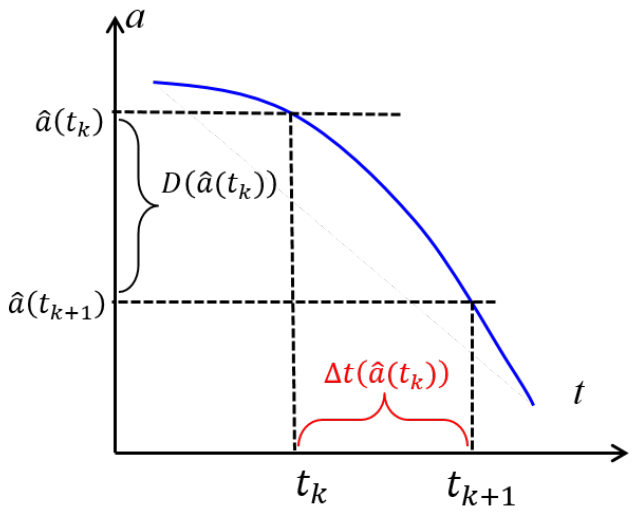


Figure 3.2: The derivation of Lebesgue fault growth model.

In order to conduct the prognosis, a model need to be developed to describe the operational time propagation at each Lebesgue state. As shown in Figure 3.2, the operation time increasing from $\hat{a}(t_k)$ to $\hat{a}(t_{k+1})$ Δt is given as $\frac{D}{\dot{a}(t_k)}$. Combining with Eq. (3.1), the operation time propagation function is approximated as:

$$t_{k+1} = t_k + \frac{D(\hat{a}(t_k))}{F(\hat{a}(t_k), u(t_k))} \tag{3.4}$$

To use this fault dynamic model in LS-FDP, we quantify the fault measurements. Lebesgue sampling basically takes samples when the difference between the current

state and the last sampled state exceeds the pre-defined Lebesgue state length. Then the LS-based model of the fault dynamics in discrete-time can be described as follows:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t(D, \dot{\hat{a}}(t_k)) \tag{3.5}$$

where $\hat{a}(t_k)$ is the Lebesgue state, t_k is the kth sampling instant, D is the Lebesgue length, and $f_t(\cdot)$ is a nonlinear function, which corresponds to the nonlinear term in (3.3).

Compared to diagnosis, prognosis requires much more computational resources mainly because of long-term predication, especially when the prediction horizon is large, which is not a rare case in FDP applications. To reduce computation time and resources, a new model is developed based on Eq. (3.4) in the LS-based prognosis as follows:

$$t_{k+1} = t_k + g_t(D, \dot{\hat{a}}(t_k)) \tag{3.6}$$

Note that $\dot{\hat{a}}(t_k) = f(\hat{a}(t_k), u(t_k))$ and $g_t(D, \dot{\hat{a}}(t_k))$ is a nonlinear function. Rather than conducting a long-term prediction on the time axis, this model calculates the RUL on each Lebesgue state directly so that the prediction horizon is the number of Lebesgue states on the fault dimension axis. Since the number of Lebesgue states on the fault dimension axis is small, the prediction horizon for LS-based prognosis is small and will significantly reduce the computation.

In this Chapter, Lebesgue state length D is manually selected as a uniform constant based on the knowledge of the system. Obviously, this selection is not optimal. In theory, the Lebesgue state length must be selected optimally and adaptively according to the fault state and fault growth rate at the current event. The optimal selection of Lebesgue state length will be one of our future works.

3.1.2 The Concept of Lebesgue Sampling

The concept of Lebesgue sampling can be illustrated through an example of degrading battery capacity [84]. The battery has a rated capacity of 1.1 Ah. The cycling of the batteries was accomplished by multiple charge-discharge tests.

Figure 3.3 shows the battery capacity in 1000 cycles. It is clear that the capacity degradation in the range $R_1 = [1,\ 780]$ cycle is slower than that in the range $R_2 = [780,\ 1000]$ cycle. Using RS-FDP with fix time interval, as shown in Figure 3.3(a), the FDP algorithms are executed at each cycle no matter if it is necessary. The setting of fix time interval, although guarantees the tracking accuracy for fault growth in range R_2, many unnecessary calculations in range R_1.

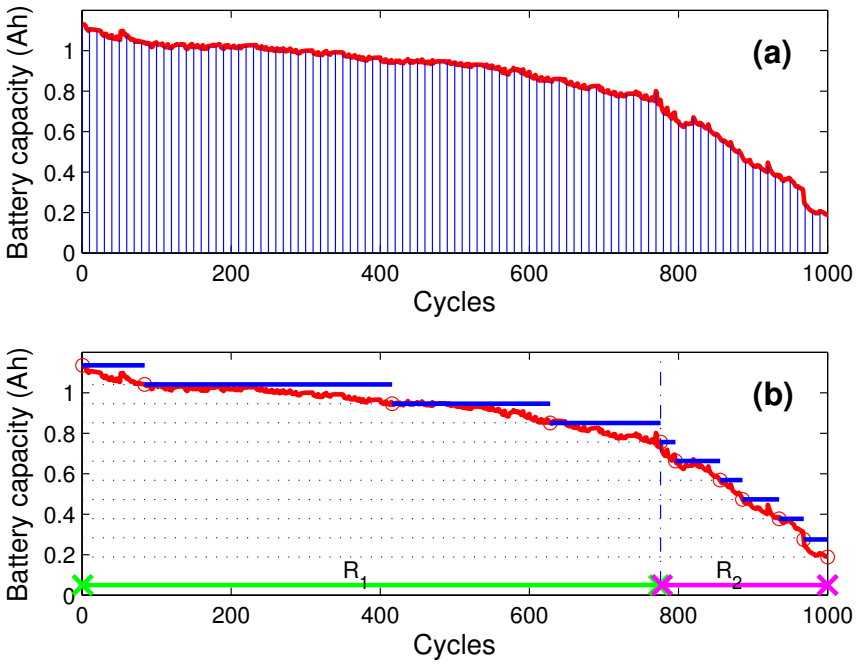


Figure 3.3: Illustration of LS. (a) RS with fixed time interval; (b) LS with fixed Lebesgue state length

Ideally, we expect to reduce the number of FDP execution in the range R_1 where the capacity degradation is slow so that more resources can be assigned to

other tasks. In the range of R_2 where the capacity degradation becomes fast, we increase the number of FDP execution by assigning more resources to FDP tasks. This setting is desirable in FPGA-based embedded systems where resources are dynamically reconfigurable and are assigned to different tasks in realtime. With this configuration, a balance between computation and performance can be achieved. This strategy however involves time-varying sampling periods that is not an easy task within the Riemann sampling framework. With Lebesgue sampling, the realization of this strategy becomes natural. By defining Lebesgue states on the vertical axis of fault dimension (battery capacity in this figure), fewer transitions between states are made when the fault growth is slow while more transitions are made when the fault growth is fast. For the example shown in Figure 3.3.(b), only 4 Lebesgue states are visited in the first 780 cycles in R_1 and 7 states in the remaining 220 cycles in R_2, which means that the FDP only needs to be executed 4 times in R_1 and 7 times during R_2. With this consideration, during R_1, more computation resources can be assigned to other tasks while only a little resources are needed for FDP. During R_2, more resources are assigned to FDP tasks so that the fault dimension can be tracked accurately.

3.1.3 Lebesgue Sampling-Based Diagnosis

In the LS-FDP framework, the range of the state $a(t)$ is partitioned into Lebesgue states $\{F_1, F_2, \cdots, F_f\}$, with which the diagnostic model is discretized. The diagnostic algorithm is executed when an event happens, *i.e.* $a(t)$ changes from one Lebesgue state to another one [85, 86]. The time instant when an event occurs is called the "event stamp". The sequence of the event stamps is denoted as $t_1, t_2, t_3, \cdots$, which formulates a time series that can be used as the input of real-time diagnostic algorithms such as a Kalman filter-based or particle filter-based algorithm [12, 81, 87–89]. The output of diagnostic algorithm is the current fault state distribution

at these event stamps and the probability of fault detection. The implementation procedure of the Lebesgue sampling-based diagnosis is illustrated in the flow charts shown in Figure 3.4.

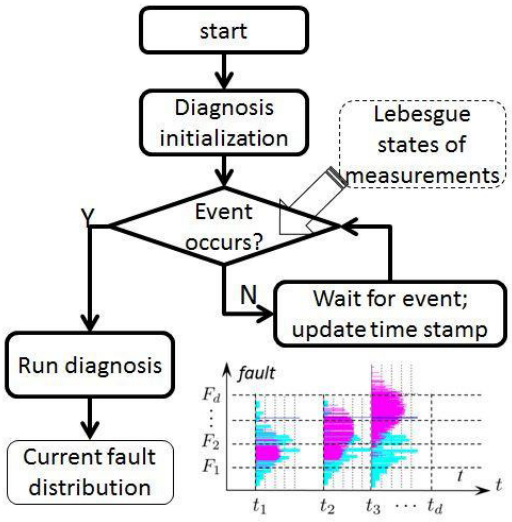


Figure 3.4: Flow chart of Lebesgue sampling-based diagnosis

3.1.4 Lebesgue Sampling-Based Prognosis

RS-based diagnosis calculates the fault state distribution at the current time instant. In the framework of Riemann sampling-based prognosis, the prediction is conducted from the current time instant $t_{current}$ to future time instants till t_{fail} when the fault state reaches a failure threshold F_f. The prognostic horizon $[t_{current}, t_{fail}]$ is usually large, especially at the early stage of the fault or when the fault growth is slow. The prediction calculates the fault state at each fixed time interval, which is demanding on the computational resources. Moreover, prognostic uncertainty will grow rapidly with large prediction horizon.

To address the challenges, a new prognostic philosophy is proposed with Lebesgue sampling. Suppose an event occurs at Lebesgue state F_d and prognosis needs to be implemented. We consider the discretized prognostic model with Lebesgue states

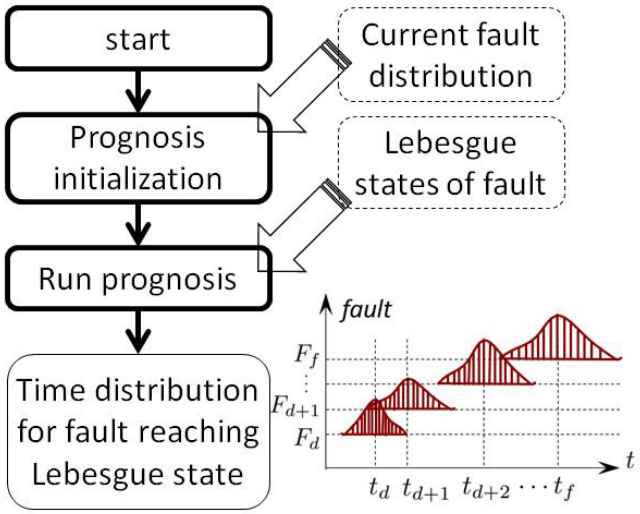


Figure 3.5: Flow chart of Lebesgue sampling-based prognosis

$\{F_d, F_{d+1}, \cdots, F_f\}$. The prognostic algorithm is implemented, together with the LS-based prognostic model, to calculate the distributions of operation time when the fault state reaches different Lebesgue states $\{F_d, F_{d+1}, \cdots, F_f\}$. Meanwhile, it will provide a RUL estimation on Lebesgue state F_f. Note that the prognostic horizon can be controlled by adjusting Lebesgue state length. Increasing the Lebesgue state length will decrease the number of events, which will reduce the required computational resources. The implementation procedure of the Lebesgue sampling-based prognosis can be illustrated in the flow charts shown in Figure 3.5.

Note that the output of diagnostic algorithm is the fault state distribution at the current time instant. For RS-based prognosis, this distribution is used as the initial condition for prognosis directly. In LS-based prognosis, the initial condition is the distribution of operation time to reach the current Lebesgue state F_d. To obtain this time distribution on F_d, a short-term prediction based on fault growth model (3.5) is conducted with the current Lebesgue state F_d being set as the threshold. Figure 3.6 illustrate the conversion process. At time t_k when an event happens on the current Lebesgue state F_d, fault diagnosis is executed and each particle is

processed to calculate the state pdf at t_k. Note that at time instant t_k, some particles does not reach the current Lebesgue state. To get the time distribution on F_d, a short-term prediction is used for those particles not yet reach F_d. This short-term prediction provide future fault state distribution, which is compared against the current Lebesgue state F_d through the law of total probability to achieve the time distribution for fault state reaching the threshold defined at F_d. This time distribution is used as the initial condition for LS-based prognosis.

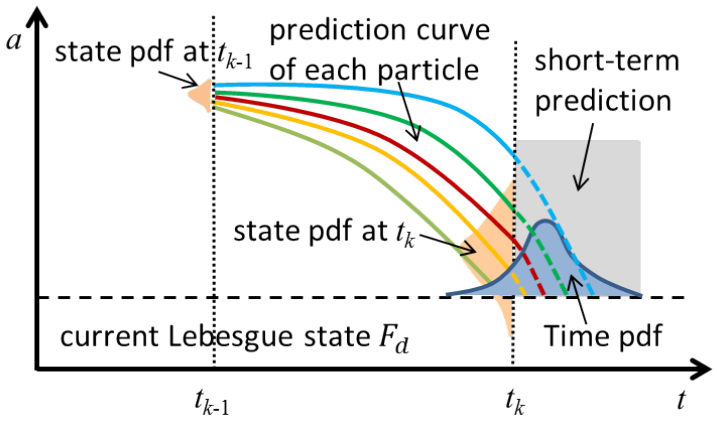


Figure 3.6: Conversion from state distribution to time distribution

3.2 Methodology Development

3.2.1 Particle Filter for LS-Based Diagnosis

The fault diagnosis is basically a state estimation problem, which can be handled in a Bayesian framework. Mathematically, assume the unobserved fault process X to be a Markov process characterized by initial distribution $p(x_0)$ and the transition probability $p(x_k|x_{k-1})$ defined by $x_k = f_k(x_{k-1}, \omega_k)$ with ω_k being the process noise. The subscript k represents the kth event stamp caused by the transition of Lebesgue states. The observations Y are assumed to be conditionally independent given X. The distribution of $(Y_k|X_k)$ is defined by $y_k = h_k(x_k, v_k)$ with v_k being observation

noise. Let $x_{0:k} = \{x_0, \cdots, x_k\}$ and $y_{1:k} = \{y_1, \cdots, y_k\}$ denote the state and the observation up to the kth event. It is of interest to estimate the *posterior distribution* $p(x_{0:k}|y_{1:k})$. The task can be achieved by two sequential steps, prediction and filtering.

In most nonlinear cases, however, analytical solutions do not exist. Alternatively, sequential Monte Carlo (SMC) methods, such as particle filter [89], provide approximate solution to state estimation that is used for fault diagnosis.

Assume that a set of N particles $(w_{k-1}^{(i)}, x_{0:k-1}^{(i)})$ is available such that they can be used to approximate a desired distribution $\pi_{k-1}(x_{0:k-1})$, where the superscript $i = 1, 2, \cdots, N$ denotes N particles located at $x_{0:k-1}^{(i)}$ and $w_{k-1}^{(i)}$ is the weight of the ith particle at the $(k-1)$th event. The objective is to efficiently obtain a new set of N particles $(w_k^{(i)}, \bar{x}_{0:k}^{(i)})$ that can approximate the distribution $\pi_k(x_{0:k})$, where $\bar{x}_{0:k}^{(i)}$ denotes location of N new particles. In the context of SMC methodology, a Monte Carlo approximation can be obtained as:

$$\pi_k(x_{0:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta\left(x_{0:k} - \bar{x}_{0:k}^{(i)}\right). \tag{3.7}$$

with $\sum_{i=1}^{N} w_k^{(i)} = 1$, where δ denotes the Dirac-delta function. The weight can be updated in a recursive formula as:

$$\begin{aligned} w\left(\bar{x}_{0:k}^{(i)}\right) &= w_{k-1}^{(i)} h_k\left(y_{1:k}|\bar{x}_{0:k}^{(i)}\right) \quad \text{and} \\ w_k^{(i)} &= \frac{w\left(\bar{x}_{0:k}^{(i)}\right)}{\sum_{i=1}^{N} w\left(\bar{x}_{0:k}^{(i)}\right)}. \end{aligned} \tag{3.8}$$

To implement the above mentioned particle filtering based fault diagnosis with LS, an LS-based diagnostic model is given by:

$$\begin{cases} x_d(t_k+1) = f_b\left(x_d(t_k) + n(t_k)\right) \\ \hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t\left(D, \dot{\hat{a}}(t_k)\right) \cdot x_d(t_k) + \omega_a(t_k) \\ y(t_k) = \hat{a}(t_k) + v(t_k) \end{cases} \tag{3.9}$$

with nonlinear mapping $f_b(x)$ is given by

$$f_b(x) = \begin{cases} 1, & \text{if} \quad \|x-1\| \leq \|x-0\| \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}$$

and the initial condition is given by $x_d = 0$, where x_d are a collection of Boolean states that indicate *normal* (0) and *faulty* (1) conditions, respectively, $\hat{a}$ is the Lebesgue state that represents the fault dimension, ω_a and v are process and observation noises, respectively, n is independent and identically distributed uniform white noise, and u is the external input. In this equation, t_k is the event stamp indicating that there is a state transition event. As assumed earlier, the feature value $y(t_k)$ indicates the fault value $\hat{a}(t_k)$ directly, in order to simplify the description. For battery case, it directly shows the measured battery capacity from Coulomb counting.

In this model, the fault diagnosis discriminates between two Boolean states, *normal* (0) and *faulty* (1), for each particle. The output of the fault diagnosis at t_k is defined as the current expectation of the Boolean state associated with the fault model for all particles, i.e. $E\{x_d(t_k)\}$. For those particles with $x_d = 1$, a faulty condition is indicated and the fault dimension progresses according to $\hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t\left(D, \dot{\hat{a}}(t_k)\right)$. While when $x_d = 0$, a health condition is indicated and the fault dimension does not progress.

3.2.2 Particle Filter for LS-Based Prognosis

In traditional RS-based prognosis, the prediction is carried out with fix time interval from the current time instant $t_{current}$ to the time instant t_{fail} that fault state reaches failure threshold F_f. The particles are estimated at each future time instant to approximate a fault state distribution at that time instant (the first prognosis level). Then, the fault distributions at all the future time instants are compared with the failure threshold F_f by applying the law of total probability to calculate the RUL distribution (the second prognosis level).

This RS-based prognostic approach often involves a large prognostic horizon, especially at the early stage of a fault and when the fault growth is slow. This large prognostic horizon causes two major issues. First, it is computationally expensive

and not suitable for applications with limited computational resource in distributed systems. Second, the uncertainty in prognosis is inherent and will accumulate as the prediction horizon increases. When the uncertainty becomes too large, the estimation of the RUL becomes unreliable that cannot be used in decision-making.

With LS, the prediction horizon reduces to the number of Lebesgue states from the current Lebesgue state F_j to failure threshold F_f. With this idea, each run of the prognostic algorithm guarantees that the fault has changed and an event has occurred. As a result, a large amount of unnecessary computation can be avoided, which is impossible with RS. It will not only reduce the requirements on computational resources, but also provide an intuitive way to manage uncertainties in prognosis.

In the context of LS, the prognostic model is given by:

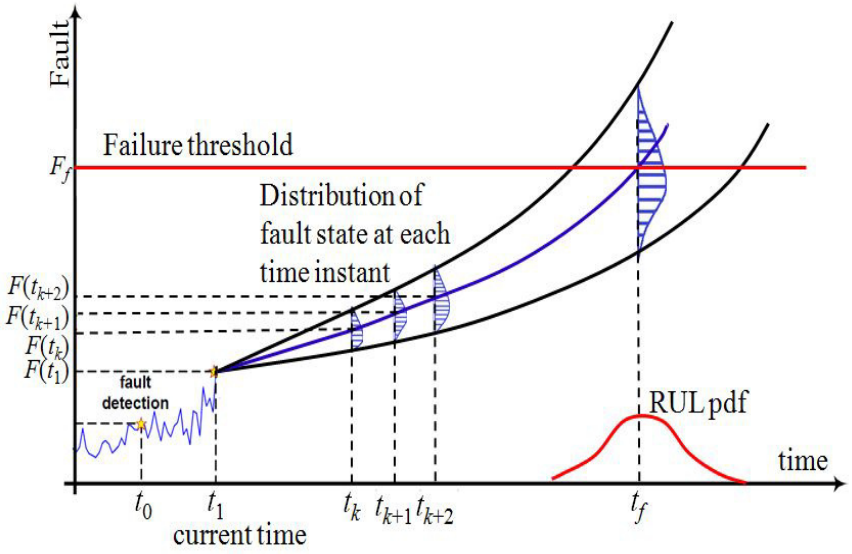
$$t_{k+1} = t_k + g_t(D, \dot{\hat{a}}(t_k)) + \omega_t(t_k) \tag{3.11}$$

where D is Lebesgue state length and $\omega_t(t_k)$ is a model noise.

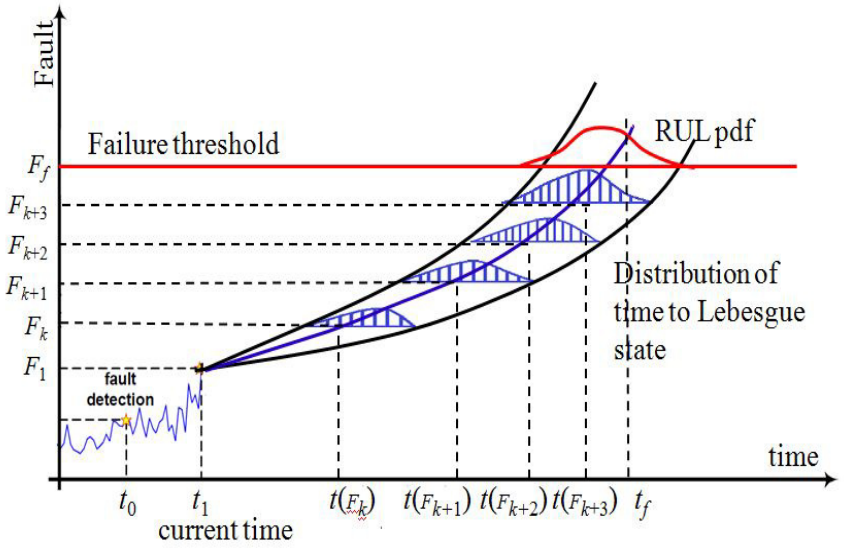
With this model, the particles are defined on the time axis instead of the fault dimension axis in RS-based prognosis. To initialize the prognosis, a new set of N particles is defined as $(w_L^{(i)}, t_L^{(i)})$, in which subscript L denotes the Lebesgue state, $w_L^{(i)}$ denotes the particle weight, and $t_L^{(i)}$ denotes particles on the time axis. The initial particles are obtained from diagnosis and short-term prediction as discussed earlier.

Note that the prognosis is carried out with a model given by (3.11). The outcome is the distributions of the operating time for the fault state to reach each Lebesgue state. Therefore, in this LS-based prognosis, the RUL pdf is calculated directly at the Lebesgue state $L = F_f$.

The difference between RS-based and LS-based prognosis is illustrated in Figure 3.7. We assume that a fault is initialized at an unknown time instant t_0. The fault is detected at t_1 and prognosis is activated from this time instant. For RS-based prognosis in Figure 3.7(a), the prediction horizon is $[t_1, t_f]$, where t_f is the time



(a) RS-based prognosis



(b) LS-based prognosis

Figure 3.7: Comparison of RS-based prognosis and LS-based prognosis

stamp when the prediction of all particles pass the failure threshold. With a sampling period of T, the prognostic algorithm needs to recursively prediction all particles $(t_f - t_1)/T$ steps and this is the most time-consuming part of prognosis which limits many applications. In other words, the prediction steps are $[t_1, \cdots, t_k, t_{k+1}, t_{k+2}, \cdots]$ on the horizontal time axis. The expectation of fault state distribution at these steps are $[F(t_1), \ \cdots, \ F(t_k), \ F(t_{k+1}), \ F(t_{k+2}), \ \cdots]$, which shows an uneven fault growth.

The process for RUL pdf prediction is shown in Figure 3.8. At first, the predicted fault state pdf at each time instant is used to compare against the failure threshold to determine how much of the fault state pdf has reached the failure threshold. This process results in a cumulative density function (cdf) of RUL and then the RUL pdf is calculated based on the cdf.

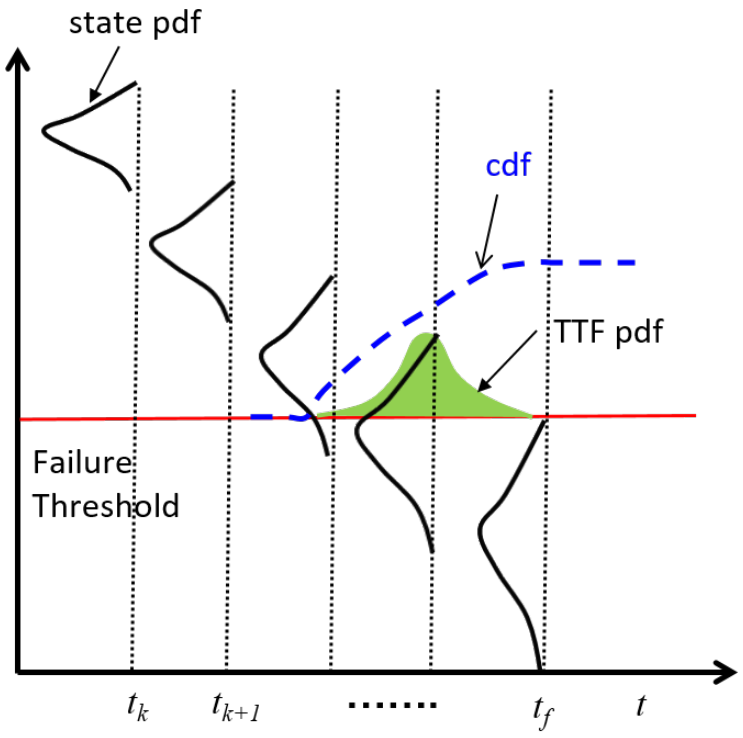


Figure 3.8: The process of RUL calculation in RS-FDP.

In the Lebesgue sampling-based prognosis, the prediction horizon is $[F_1,\ F_f]$ where F_f is the fault dimension that indicates the failure of the system. With a uniform Lebesgue length of D, there will be $(F_f - F_1)/D$ predication steps, and can be denoted as $[F_1, \cdots, F_k, F_{k+1}, \cdots, F_f]$ on the vertical axis. The expectations of the distributions of the operating time for the fault reaching these Lebesgue states are $[t_1, \cdots, t(F_k), t(F_{k+1}), \cdots, t_f]$, of which the time intervals are uneven. In summary, the fundamental difference is that RS-based prognosis calculates fault state distribution at given time instants, while LS-based prognosis calculates time

distribution at predefined Lebesgue states.

3.3 Experimental Results

In this experiment, the SOH of a Lithium-ion battery with rated capacity of 1.1 Ah is used to verify the proposed LS-FDP algorithm. The charge-discharge cycle of the battery is conducted by Arbin BT2000 system under room temperature at a discharge current of 1.1 A [90]. The battery capacity degradation speed is the decrease of the capacity with respect to charging-discharging cycles. The capacity degradation curve vs charging-discharging cycle is obtain by Coulomb counting. The charge-discharge cycle is cut off at pre-determined cut-off voltages of 2.7 V for discharging and 4.2 V for charging, respectively. The battery capacity degradation data are shown in Figure 3.9. The failure threshold is defined as 0.25 Ah and the corresponding battery capacity degrades to this value at 854th cycle for CS2-36 battery.

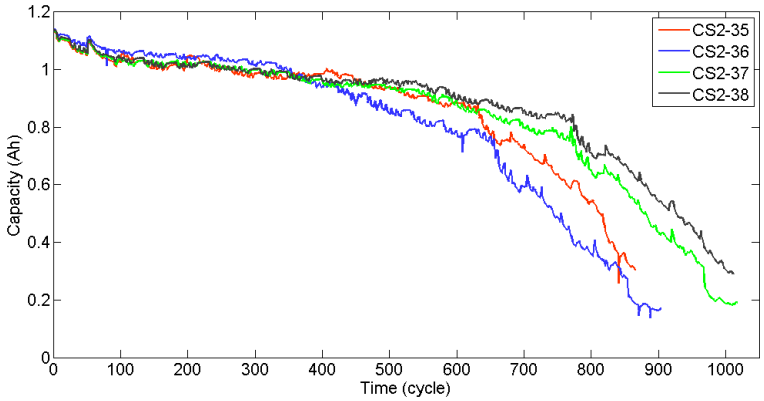


Figure 3.9: Battery capacity degradation data vs. charging/discharging cycle.

3.3.1 RS-FDP

To implement diagnosis and prognosis, a fault growth model needs to be developed. For RS-FDP, the fault growth model is given by:

$$C(t+1) = C(t) - \gamma \cdot \left| p_1 \cdot \left(p_2 + p_3 \cdot t + p_4 \cdot t^2 \right) \right|^{p_5} + \omega(t) \tag{3.12}$$

where C is battery capacity, t is the time index indicating the charging-discharging cycles, $p = [5e^{-5}, \ -215, \ 4.8, \ -0.0135, \ 0.4]$ are parameters, γ is a hyper model parameter with mean of $3.8e^{-3}$ and variance of $5e^{-}5$, and ω is a model noise with 0 mean and $1e^{-}6$ covariance.

For RS-based diagnosis, a particle filter with 500 particles is used. Since the capacity degrades as soon as the battery was put into use, our focus is to estimate the capacity state. Figure 3.10 shows the diagnostic results at the 472nd cycle. The expected value of battery capacity is 0.8756 and the 95% confidence interval is [0.8486, 0.8851]. In this figure, the top subfigure is the measurement (given by capacity from Arbin system's Coulomb counting) compared with the filtered estimation (given by the expected value of estimation distribution). The bottom subfigure shows the comparison of initial baseline pdf (at the beginning of battery use without capacity degradation) with the real-time estimation pdf at the 472nd cycle. Note that in this RS-based diagnosis, the diagnostic algorithm needs to execute 472 times, *i.e.*, every time when a new measurement becomes available.

With an estimation of the current battery capacity as the initial condition, prognostic algorithm is implemented to conduct the long-term predication and estimation of RUL. Figure 3.11 shows the prognosis of battery capacity degradation with the RUL estimation. To make the figure clear, the battery capacity distribution at each cycle is not shown. Instead, this figure shows the expected value, upper and lower bound of 95% confidence interval of the battery capacity pdf at each future cycle. Note that the prognosis needs to predict all particles from its current value at the cycle 472 to the failure threshold value. In this figure, the prediction horizon is 526 cycles. To make the real-time implementation of prognosis possible, the number of particles is reduced to 20.

Then, the battery capacity pdf at each cycle is compared with the failure threshold to obtain the RUL pdf, as shown in the histogram on the horizontal axis. This process

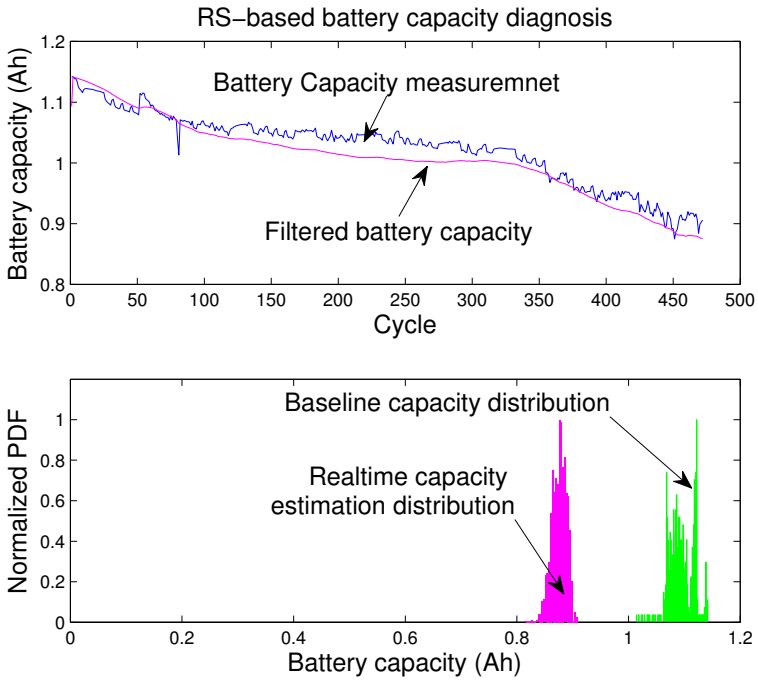


Figure 3.10: Experimental result of RS-based diagnosis.

uses the law of total probabilities and can be mathematically described as:

$$p_{failure}(t) = \sum_{i=1}^{N} Pr\left(Failure|x_t^{(i)} > F_f\right) w_t^{(i)} \tag{3.13}$$

where superscript (i) is the index of particles, $p_{failure}(t)$ is the probability of failure at time t, $w_t^{(i)} = Pr(x = x^{(i)})$ is the weight of particles at time t, and x_t is the predicted value of a particle at time t.

In this figure, the predicted expectation of the failure time is at the 763.3 cycle and the RUL life is 291.3 cycles. The distance from the predicted expected value to the ground truth value is 90.7 cycles. The 95% confidence bound of the RUL pdf is given as [636 992], which shows that the uncertainty caused by the long prediction horizon is very large.

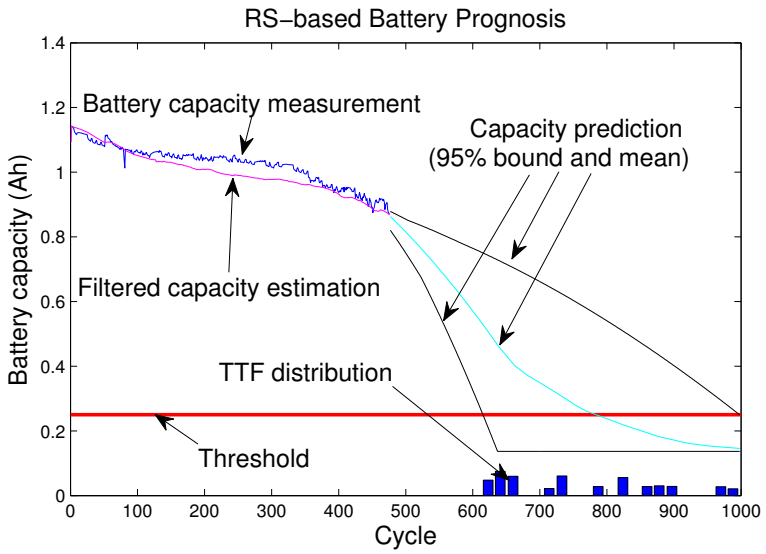


Figure 3.11: Experimental result of RS-based prognosis.

3.3.2 LS-based FDP

To implement the LS-FDP, 40 Lebesgue states are defined in the battery's full capacity of 1.1Ah. With this setting, the diagnostic algorithm is executed only when the capacity degrades from one Lebesgue state to another, *i.e.*, an event happens.

The models for diagnosis and prognosis are given by,

$$C(t_{k+1}) = C(t_k) - p_d \cdot D \cdot sgn\left(C(t_k) - C(t_{k-1})\right) + \omega_C(t_k) \tag{3.14}$$

where $p_d = 1.1$ is model parameter, D is Lebesgue state length, t_k is event stamp indexes, $sgn(\cdot)$ gives the sign as discussed in (3.2), and ω_C is model noise.

Figure 3.12 shows the diagnostic results at the 472nd cycle. The particle filtering algorithm uses 500 particles. The expected value of battery capacity is 0.9026 and the 95% confidence interval is [0.8439, 0.9696]. Same as the RS-FDP case, the top subfigure is the capacity measurement from Coulomb counting compared with the filtered capacity estimation. Note that the flat segments mean no event and the diagnostic algorithm does not execute. The lower subfigure shows comparison of the initial baseline pdf against the realtime estimation pdf at the 472nd cycle. During these 472 cycles, there are 76 events, *i.e.*, the diagnostic algorithm only runs 76

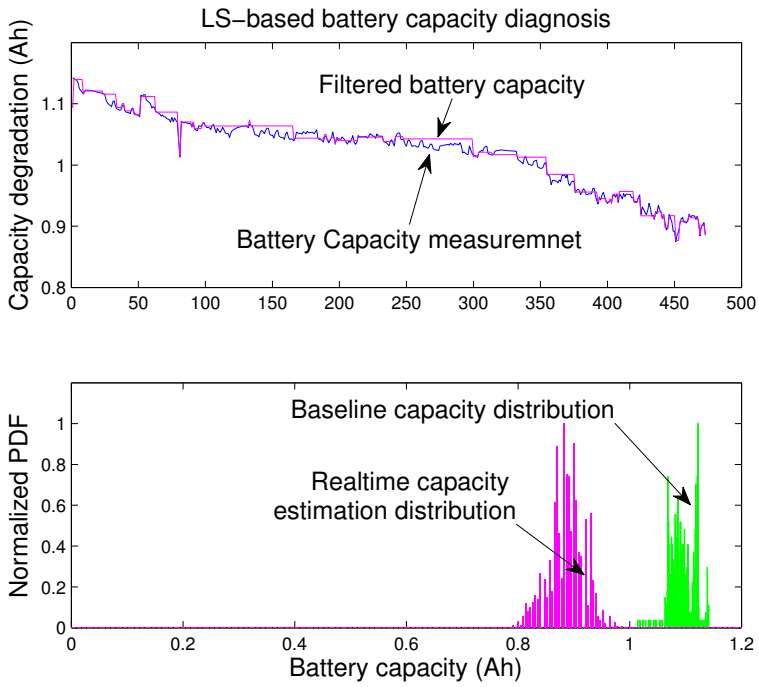


Figure 3.12: Battery result of LS-based diagnosis at the 472nd cycle

times. Therefore, the LS-based diagnostic algorithm only runs 76 times. Compared with traditional RS-FDP algorithm that needs to run 472 time, the reduction of computation is (472-76)/472=83.9% and computation is 6.21 times faster, which is a remarkable improvement.

Different from RS-FDP in which prognosis runs at every cycle, LS-based prognosis is only executed along with diagnosis. That is, diagnosis and prognosis are executed only when an event happens. To conduct LS-based prognosis, the operation time distribution at the current Lebesgue state is projected to all future Lebesgue states, as shown in Figure 3.7(b). Since the LS-based diagnosis outputs the estimation pdf of battery capacity, the short-term prediction based on (3.14) is used on those particle not yet reach the current Lebesgue state. Then (3.13) is used to get the initial time distribution for prognosis.

Unlike RS-FDP, in which the diagnostic and prognostic model are essentially the same physical fault growth dynamics, the LS-based prognosis is conducted on fault dimension axis to predict the time-to-Lebesgue-state directly. The diagnostic model

(3.14) cannot be used in prognosis as we discussed in Section 3.1. The prognostic model used in LS-based prognosis is given as:

$$t_{k+1} = t_k + p_p \cdot D \cdot exp\left(-\dot{C}(t_k)\right) + \omega_k(t_k) \tag{3.15}$$

in which $p_p = 0.8$ is model parameter and ω_k is a model noise.

Figure 3.13 shows the prognostic results at the 472nd cycle. In this figure, the prediction horizon is only 24 Lebesgue states, which is very small compare to the 526 cycles in RS-FDP. The LS-FDP prognosis can afford the computation of 500 particles and we do not need to reduce the number of particles.

To make the figure clear, only the time distribution pdf at a few selected Lebesgue state are plotted. Note that the time distribution pdf at the Lebesgue state defined by the failure threshold gives the RUL estimation pdf. In this figure, the predicted failure time is at the 859.9 cycle and the RUL life is 387.9 cycles. The 95% confidence bound of the RUL pdf is given as [744.7 913.9]. The uncertainty is much smaller than that of RS-based prognosis. When the predicted time to failure pdf expected value compared with the ground truth value of 854 cycle, the difference between them is 4.9 cycles.

Compared to RS-based prognosis with large horizon (526 cycles) and small number of particles (20), the LS-based prognosis shown in Figure 3.13 only has a prognostic horizon of 24 Lebesgue states and can afford 500 particles. The reduction of computation time is (2.231299-0.010838)/2.231299=99.51% and the computation is about 206 times faster. The overall speed of LS-FDP is $6.21 \times 206 = 1279.2$ times faster than RS-FDP. Note that this comparison is based on the fact that only 20 particles (at the cost of performance) are used in RS-based prognosis to make real-time implementation possible.

The comparison of LS-FDP and traditional RS-FDP of battery at the 472nd cycle is summarized in Table 3.1. Note that LS-FDP offers better performance than RS-FDP in terms of time to failure (TTF) prediction due to short prognostic horizon.

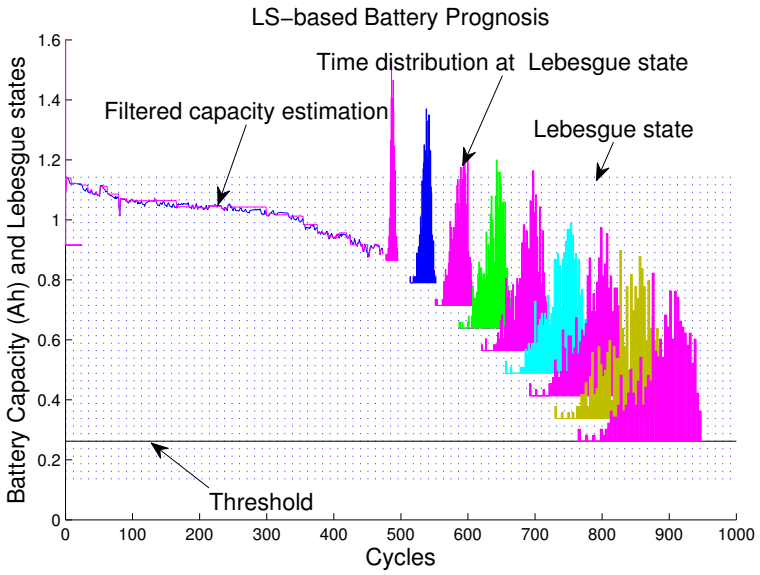


Figure 3.13: LS-based prognosis at the 472nd cycle

Table 3.1: Comparison of RS-FDP and LS-FDP results for battery (CS2-36)

	RS-FDP	LS-FDP
Diag. particles	500	500
Capa. expectation	0.8756	0.9026
Capa. 95% CI	[0.8486, 0.8851]	[0.8439, 0.9696]
Exec. numbers	472 (100%))	76 (16.1%)
Prog. particles	20	500
TTF expectation	763.3	859.9
TTF 95% CI	[636, 992]	[744.7, 913.9]
Prog horizon	526	24
Comp. time (s)	2.231299 (100%)	0.010838 (0.49%)
Overall speed	1	$6.21 \times 206 = 1279.2$

The advantages of Lebesgue sampling in fault diagnosis and prognosis are obvious from the comparison of above experimental results. For the diagnosis, the two approaches show the comparable performance. In terms of prognosis, the LS-FDP shows better performance in terms of accuracy and precision. First, Lebesgue sampling in FDP greatly reduces the computation time and the requirement of computation resources without sacrificing the performance of diagnosis. Since prognosis in Riemann sampling framework usually have a large prediction horizon, it often needs more computation time and resources. This in consequence becomes

a main limitation of prognosis for those applications with fault tolerant control and reconfigurable control, where the real-time calculation of RUL is critical. Another important issue with large prediction horizon in Riemann sampling is the significant accumulation of uncertainties in prognosis and the degradation of the performance of prognosis in terms of accuracy and precision. The introduction of Lebesgue sampling in FDP provides a natural solution for real-time implementation, especially on those systems (such as embedded systems) with limited computation capability. The prediction horizon of LS-FDP can be very small comparing to that of RS-FDP, this is very good in managing the uncertainties in prognosis.

Accuracy is one of the most important properties in FDP. In order to compare the result of RS-FDP and LS-FDP methods, α - λ matrix is introduces in [83] as discussed in Chapter 1. It is re-written here for convenience, the matrix is defined as:

$$[1-\alpha]\cdot r_t(t_k) \leq r^l(t_k) \leq [1+\alpha]\cdot r_t(t_k) \tag{3.16}$$

where r^l is the predicted RUL at the lth time instant, r_t is the ground truth TTF, α is the accuracy modifier [83].

Figure 3.14 shows the α-λ metrics with α=0.3 for four sets of battery capacity degradation data. Because of the short prediction horizon and small uncertainty accumulation, the prediction accuracy for LS-FDP is higher than that of RS-FDP. The mean RUL of LS-FDP reaches the accuracy zone quickly, and the prediction results are stable. The result of RS-FDP exceeds the accuracy limits more often, which means the estimation of RUL is not uniformly accurate, and the 95% CI is much bigger than that of LS-FDP, as shown in Table 3.1. The high prediction accuracy of LS-FDP is achieved with much lower computation cost compared with RS-FDP.

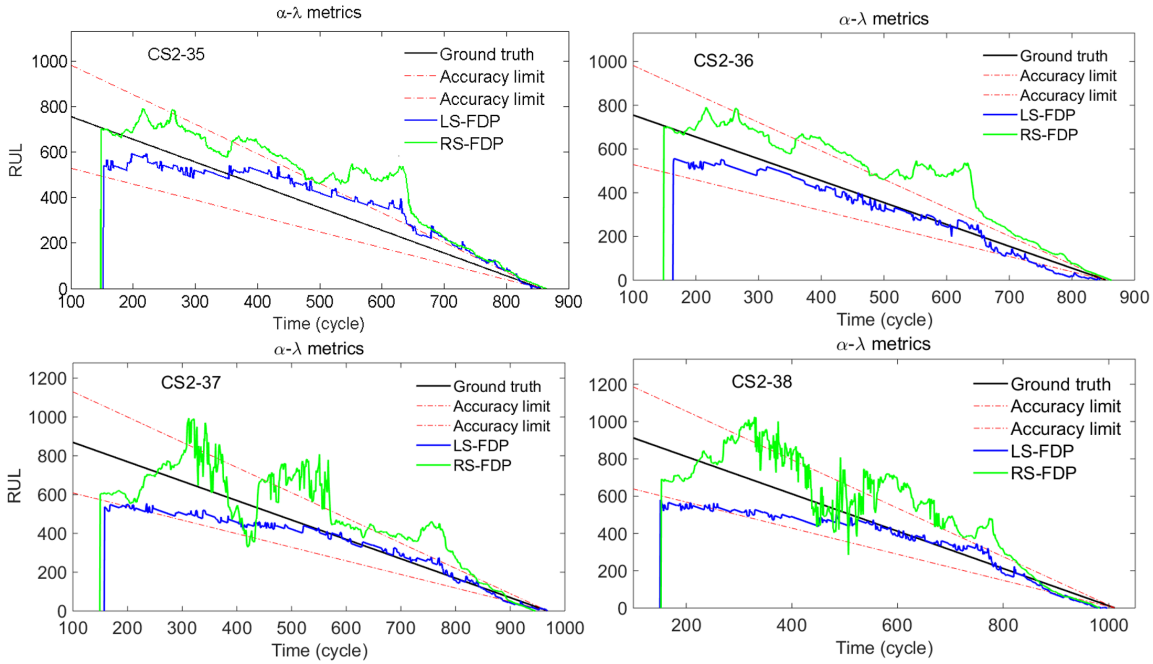


Figure 3.14: Prognostic accuracy comparison for RS-FDP and LS-FDP

3.4 Conclusion

The Chapter introduces the concept of Lebesgue sampling into FDP and develops a novel Lebesgue sampling-based FDP approach with an philosophy of "execution only when necessary" or an "as-needed" basis. With this new feature, the LS-FDP enables the FDP on systems with limited computation capabilities and enables the distributed FDP. The contributions of this work include: 1). Propose a new LS-FDP framework with a complete design methodology, which is able to accommodate different diagnostic and prognostic algorithms; 2) Develop a particle filtering-based FDP algorithm in the LS-FDP framework; and 3) Present an experiment of Lithium-ion battery SOH diagnosis and prognosis with comparison against traditional RS-based approach. It is worth mentioning that the proposed LS-FDP requires to design diagnostic model and prognostic model separately. The reason behind this is that LS-based diagnosis is based on the growth of fault dimension while LS-based prognosis is based on the calculation of operation time to reach different Lebesgue states defined

as different fault dimensions. Experimental comparison against RS-FDP shows that the proposed LS-FDP is able to reduce the overall computation time significantly without sacrificing the performance.

Chapter 4

Uncertainty Management in LS-FDP

The distributions of fault state estimation and time to failure (TTF), obtained from FDP algorithms, are influenced by the uncertainties originate from inaccuracy model, measurement noise, process noise, and unknown future loading, which make it challenging for decision-making. In order to arrive at a trust-worthy TTF, the uncertainties need to be analyzed and various sources of uncertainty related to FDP must be taken into account.

Models for FDP need to be accurate to ensure the accuracy of the prediction of RUL. However, due to various factors including the lack of understanding of the fault dynamics, the uncertainties from future loading, and measurement noise, a model with fixed parameters is insufficent to accurately describe the effects of varying operational and environmental conditions on fault growth and will result in performance degradation. One of the solutions to overcome this problem is to adopt the model parameter adaptation in the algorithms, which adjusts the parameters in the model adaptively based on new measurements. The parameter adaptation has been studied in traditional RS-FDP, in which the samples are taken in a periodic manner. Since FDP method based on Lebesgue sampling (LS) has been developed in Chapter 3 [91, 92] to overcome the high demands on computational resources, it is necessary to study the parameter adaptation in LS-FDP.

The measurement noise and process noise are usually represented by the noise terms in the diagnosis and prognosis models, which need to be managed to improve the performance of the algorithms. The FDP results produced by the FDP models

with noise term could show big discrepancy with the real applications. In this case, the noise term needs to be adjusted adaptively according to the measurements to achieve accurate and precise state estimation. If all the uncertainty sources are managed correctly, the state estimation and RUL prediction will converge to the true distributions.

Noise term adjustment in tradition Riemann sampling-based FDP (RS-FDP) has been studied and different methods have been developed to quantify and manage the uncertainties in TTF prediction [93–97]. However, uncertainty has not been studied in Lebesgue sampling-based FDP (LS-FDP), which motivates our research of uncertainty management in LS-FDP in this Chapter.

4.1 Model for LS-FDP

The model described by Eq. (3.9) and (3.11) is derived from the Lebesgue sampling theory. However, the predicted results show discrepancy with the real fault state and real RUL because of uncertainties in the fault growth dynamic and operation conditions. Thus, a new model is developed to address this problem, which includes the fault state, and is described as follows:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t(p_d, D, \hat{a}(t_k), \dot{\hat{a}}(t_k)) + \omega_a(t_k) \tag{4.1}$$

where p_d is a parameter introduced, $\hat{a}(t_k)$ is the fault size at t_k time instant of the k-th event, D is a set of Lebesgue state length at t_k, which is defined as the distance between the neighboring Lebesgue states: $D_i = F_{i+1} - F_i$, $\dot{\hat{a}}(t_k)$ is the degradation speed, $\omega_a(t_k)$ is the noise, and $f_t(\cdot)$ is a nonlinear function that represents the fault growth.

The prognosis model (3.11) in Chapter 3 also shows discrepancy with the ground truth value in the RUL prediction, thus a new model for LS-based prognosis is given

as:

$$t_{k+1} = t_k + g_t(p_p, D, \hat{a}(t_k), \dot{\hat{a}}(t_k)) + \omega_t(t_k) \tag{4.2}$$

where p_p is a parameter introduced for uncertainty management, $\omega_t(t_k)$ represents the uncertainties.

Note that the output of diagnosis is a fault state distribution defined on the state axis, which cannot be used in LS-based prognosis. The LS-based prognosis needs an initial condition of time distributions on the current Lebesgue state F_d. The pdf conversion is discussed in Figure 3.6.

4.2 Parameter adaptation and Noise management

4.2.1 Parameter adaptation

In the design of FDP, models for diagnosis and prognosis are needed to estimate the current health state and predict the RUL. To make the models be adaptive to changing operating conditions and environmental factors that affect the system degradation, model parameter adaptation is adopted in the FDP process. Several optimization methods have been developed, among which recursive least square (RLS) with a forgetting factor is employed due to its simplicity. The design of RLS in LS-FDP is described as follows:

1. Define a cost function as:

$$J(\theta) = \frac{1}{2} \cdot \sum_{i=1}^{N} \lambda^{N-i} \left[C(i) - C\left(\hat{\theta}(i-1)\right) \right]^2 \tag{4.3}$$

   where N is the length of the measurements at which it causes an event, λ is a forgetting factor, which is usually given in the range of $0 < \lambda \leq 1$, and $\hat{\theta}(i)$ is the model parameters.

2. Calculate the derivatives $\phi(i)$ with respect to parameters θ:

$$\phi(i) = \left.\frac{dC(\theta)}{d\theta}\right|_{\theta=\hat{\theta}(i-1)} \tag{4.4}$$

3. Parameters are updated as:

$$\hat{\theta}(i) = \hat{\theta}(i-1) + P(i)\phi(i)\left[C(i) - C\left(\hat{\theta}(i-1)\right)\right] \tag{4.5}$$

and $P(i)$ is given as:

$$P(i) = \frac{P(i-1)}{\lambda}\left[1 - \frac{\phi(i)\phi^T(i)P(i-1)}{\lambda + \phi^T(i)P(i-1)\phi(i)}\right] \tag{4.6}$$

4.2.1.1 Parameter adjustment in diagnosis

The flow chart of RLS based optimized method in diagnosis is shown in Figure 4.1. The LS-based diagnosis is executed with initial diagnosis model parameters when the measurement reaches a new Lebesgue state and it calculates a posteriori state pdf. The error between the measurement and mean of the state estimation is used as the input of the parameter adaptation algorithm to adjust the parameters based on RLS method. The updated parameter is used in the next execution of the diagnosis process. This procedure is conducted iteratively to obtain an accurate state estimation.

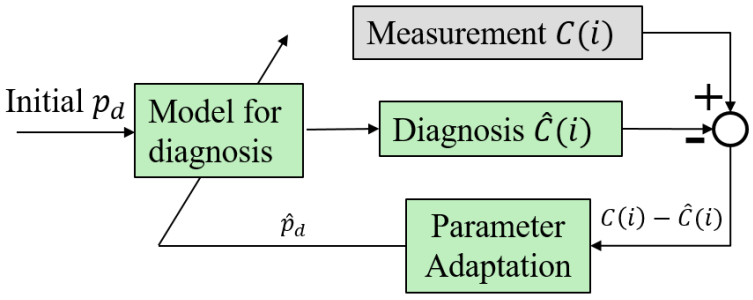


Figure 4.1: Parameter adaptation in diagnosis.

4.2.1.2 Parameter adjustment in prognosis

Parameter adaptation in prognosis is illustrated in Figure 4.2. The details are listed as follows:

1. At the current Lebesgue state F_C, prognostic algorithm is executed with the initial parameters. The time distribution for battery capacity degrading to the future Lebesgue state F_{C+p} is calculated and stored as $\hat{t}_{C+p}$, where the value of p indicates the horizon of prediction and is problem dependent.

2. When the measurement reaches F_{C+p}, the diagnosis is executed and a state distribution is achieved.

3. The state distribution is converted to a time distribution t_{C+p}.

4. The difference of the mean values of $\hat{t}_{C+p}$ and t_{C+p} given by equation (4.7):

$$e_{t,C+p} = t_{C+p} - \hat{t}_{C+p} \tag{4.7}$$

   is used in the RLS optimization, which will generate a new set of parameters to be used in the next prognosis process.

5. Repeat the steps above to continuously optimize the parameters in FDP.

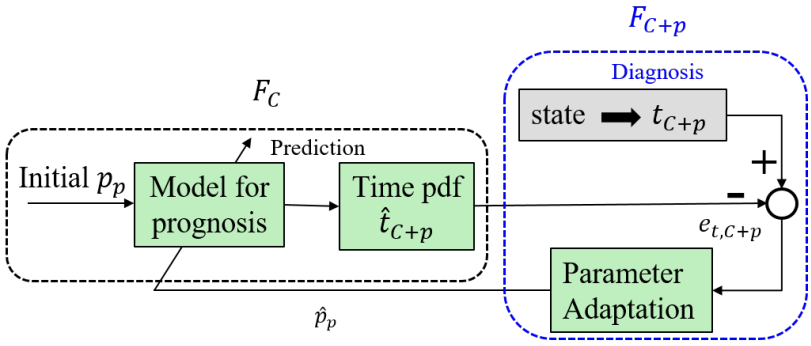


Figure 4.2: Parameter adaptation in prognosis.

4.2.2 Noise adjustment

Based on model (4.2), prognostic algorithm predicts the time distribution for fault reaching each Lebesgue state. In prognosis, the uncertainties are represented by the model noise $\omega_t(t_k)$. This section will discuss the online adjustment of $\omega_t(t_k)$ to reduce the uncertainties in RUL prediction.

In particle filtering-based framework, the proposed noise adjustment follows the procedure as described in Figure 4.3. Initially, the diagnosis algorithm is conducted at t_k when an event happens at the Lebesgue sate F_d. The prognosis is triggered at the Lebesgue state F_d. Based on model (4.2), the time distribution at the $(d+m)$-th Lebesgue state F_{d+m}, denoted as $\hat{t}_{d+m}$ can be calculated and stored. Note that the distribution of $\hat{t}_{d+m}$ is based on the prognostic noise terms at F_d and is a priori distribution from prediction. Here m is a predefined length of noise adjustment prediction horizon, which is often small. That is, the prognostic algorithm predicts the operation time distribution at the Lebesgue state F_{d+m}. When the measurement triggers an event at Lebesgue state F_{d+m}, the diagnosis is executed and the results are converted into time distribution as described in Figure 3.6. This yields a posteriori time distribution t_{d+m}. Then the stored predicted distribution $\hat{t}_{d+m}$ and measurement updated distribution t_{d+m} are compared in terms of variance to adjust the variance of the noise terms. If the variance of the stored priori $\hat{t}_{d+m}$ is larger than that of the posteriori t_{d+m}, the variance of the noise $\omega_t(t_k)$ in the prognosis model (4.2) will be decreased. Otherwise, it will be increased.

The adjustment of the variance of the prognosis noise in model (4.2) is summarized as follows:

$$\omega_t^{new}(t_k) = \frac{\sigma^2_{t_{d+m}}}{\sigma^2_{\hat{t}_{d+m}}} \cdot \omega_t^{old}(t_k) \tag{4.8}$$

where $\sigma_{t_{d+m}}$ and $\sigma_{\hat{t}_{d+m}}$ are the variance of the posteriori distribution t_{d+m} and predicted priori distribution $\hat{t}_{d+m}$, respectively. Note that the stored time distribution

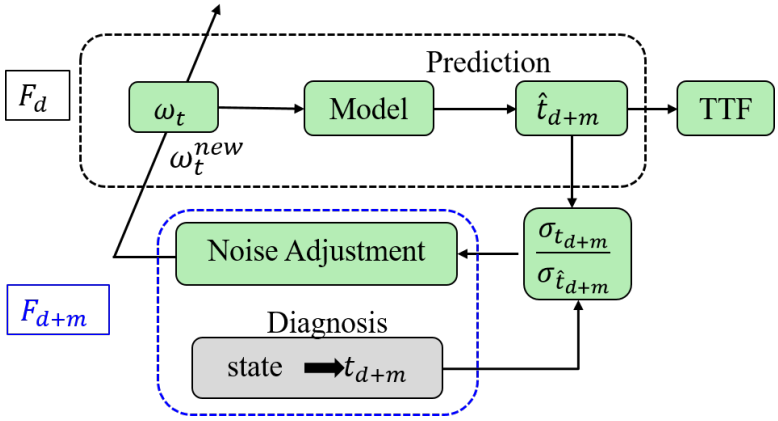


Figure 4.3: Noise adjustment in prognosis

$\hat{t}_{d+m}$ will not be updated in prognosis process. When the measurement triggers an event at F_{d+m}, the time distribution at F_{d+m} is calculated as t_{d+m} and it will be different from $\hat{t}_{d+m}$. This enables the adjustment of noise terms in the model to manage the uncertainties in prognosis.

4.3 Application with SOH prediction of Lithium-ion battery

In this section, the proposed parameter adaptation and noise adjustment methods are demonstrated with an application to the capacity degradation of Lithium-ion batteries in particle filtering. The implementation of the diagnosis algorithm in PF is discussed in Section 3.2.1. The results are compared against those from PF without parameter adaptation/noise adjustment to illustrate the advantages.

For online application, when a battery is put into service and its SOH is monitored, its current, voltage, and sampling time are recorded. The capacity is calculated by the Coulomb-counting method and is filtered by the proposed method to achieve accurate capacity estimate. The capacity approximation from Coulomb-counting method is compared with the predefined Lebesgue states to check if it triggers an event. If an event happens, diagnostic algorithm is conducted to calculate the capacity based on Bayesian theory. The capacity distribution is then converted into a time distribution

(Fig. 3.6), which is used as the initial condition of prognosis. The prognosis is executed to calculate time distributions when the capacity reaches each Lebesgue state. The time pdf at the Lebesgue state whose value equals to the failure threshold is the predicted time to failure distribution.

In this experiment, the proposed approach is tested on SOH of a set of four Lithium-ion batteries (denoted as CS2-35 to CS2-38) with 1.1 Ah rated capacity under 1 C charge/discharge rate. The details of battery degradation data is given in Section 3.3.

4.3.1 Diagnosis and prognosis models for lithium-ion battery

4.3.1.1 Diagnosis model

To implement diagnosis based on PF method, 40 uniformly distributed Lebesgue states are initially defined in the battery's full capacity of 1.1 Ah. With this setting, the diagnostic algorithm is executed only when the capacity degrades from one Lebesgue state to another. Since the diagnosis and prognosis in LS based on Eq. (3.14) show discrepancy with the ground truth RUL. A new fault growth model is developed, and is given as follows:

$$C(t_{k+1}) = C(t_k) - p_d \cdot C(t_k) \cdot D \cdot sgn(C(t_k) - C(t_{k-1})) + \omega_C(t_k) \tag{4.9}$$

where C is the battery capacity, p_d is the model parameter, t_k is the event stamp indexes when an event happens, $sgn(\cdot)$ gives the sign, and ω_C is the model noise.

4.3.1.2 Prognosis model

To conduction prognosis in LS framework, the model for LS-based prognosis is developed based on model (3.15) and given as [92]:

$$t_{k+1} = t_k + p_p \cdot C(t_k) \cdot D \cdot exp\left(-\dot{C}(t_k)\right) + \omega_t(t_k) \tag{4.10}$$

where p_p is the model parameter and ω_t is the model noise.

The procedures of diagnosis and prognosis are illustrated in Fig. 4.4. The estimation of the capacity and RUL estimation are based on the Bayesian theory, which includes two steps: prediction and correction. The battery degradation is described as Eq. (4.9). The capacity distribution at t_k-th event is approximated by a set of particles (w_i, x_i) , where w_i and x_i are the weight and location of the particles. The prior capacity distribution at t_{k+1} is predicted by propagating the particles based on Eq. (4.9). When the measured capacity is available for the t_{k+1}-th event, the posterior capacity distribution is calculated by Bayesian theory, the capacity estimation is achieved.

The RUL prediction contains two step. The first step is: the posterior capacity distribution from diagnosis at t_k-th event is converted into a time distribution. Secondly, the achieved time distribution is in step 1 used as the initial condition in prognosis to predict the time distributions for fault state reaching each Lebesgue state based on Eq. (4.10). The time distribution on the Lebesgue state defined on the failure threshold is the RUL pdf.

4.3.2 Experimental results comparison

4.3.2.1 Efficiency of LS-FDP compared with LS-FDP

The efficiency of the proposed LS-FDP is compared with that of RS-FDP in terms of number of algorithm execution and execution time, three snapshots at the 400th, 500th, and 600th cycle of the experimental results are summarized in Table 4.1 in this document. Take the execution at the 400th cycle of battery CS2-36 as an example. For LS-based diagnosis, there are 53 events in these 400 cycles and therefore it only

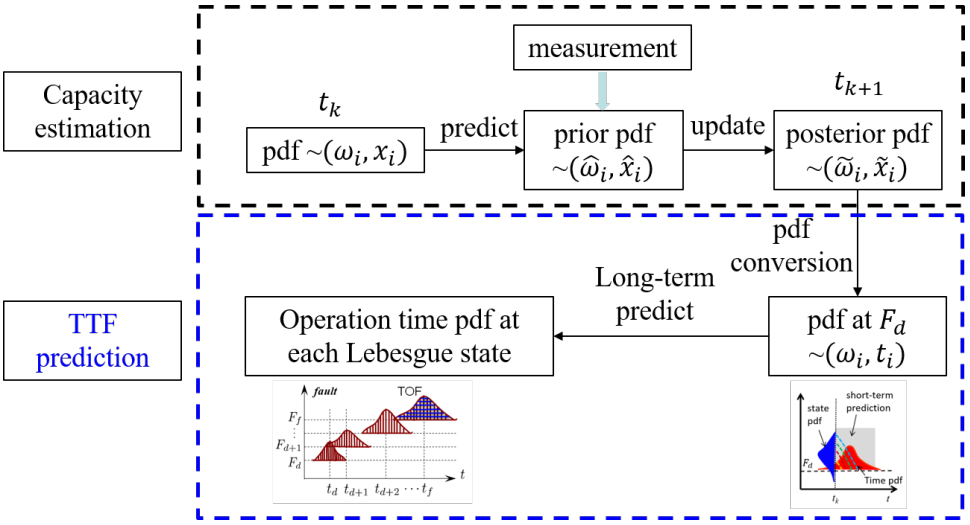


Figure 4.4: The procedure of capacity estimation and TTF prediction.

runs 53 times while RS-FDP needs to run 400 times. The reduction of execution is 86.8%. When prognosis is conducted, the prediction horizon of LS-FDP is 56 Lebesgue states while the prediction horizon of RS-FDP is 764 cycles. The execution time of LS-FDP is 0.0077 seconds with 500 particles while that of RS-FDP is 2.346 second with only 20 particles. The reduction of execution time is 99.7%. Note that for LS-FDP, when diagnosis is not executed, prognosis is not executed. Therefore, the proposed LS-FDP is approximately 1692 times faster than RS-FDP.

Table 4.1: Comparison of the LS-FDP and RS-FDP results

LS-FDP	400th cycle	500th cycle	600th cycle
Diagnosis executions	53	91	107
Particles in diagnosis	500	500	500
Prognosis time (s)	0.007726	0.006833	0.006831
Particles in prognosis	500	500	500
Execution numbers	56	27	20

RS-FDP	400th cycle	500th cycle	600th cycle
Diagnosis executions	400	500	600
Particles in diagnosis	500	500	500
Prognosis time (s)	2.3467	1.4657	1.4407
Particles in prognosis	20	20	20
Prediction horizon	764	486	473

4.3.2.2 Diagnosis results

For the diagnosis model, p_d is the target parameter to be optimized. The optimization follows the method described in Section 4.2.1.

Battery CS2-36 is chosen to illustrate the details of the experiment. Figure 4.5 shows the diagnostic results with/without parameter adaptation at the 400th cycle. The particle filtering algorithm uses 500 particles to approximate the state distribution. The subfigure (a) shows the comparison of capacity from Coulomb counting (blue) against the estimated mean value from diagnosis with/without (magenta/black) parameter adaptation. As shown in Figure 4.5 (a), the mean of the diagnosis result from the algorithm with parameter adaptation shows that a more accurate state estimation. Note that the flat magenta segments mean no event and diagnosis is not executed since the measurements do not reach a new Lebesgue state.

The diagnosis is initially executed with $p_d = 1.2$, which is changed to $p_d = 1.215$ at the 400th cycle. The expected value of battery capacity with and without parameter adaptation are 0.9476 and 0.9534, respectively, and the corresponding 95% confidence intervals are $[0.8872,\ 1.0054]$ and $[0.8930,\ 1.0112]$, respectively.

The real-time state distribution against the baseline distributions at the 400th cycle for the algorithm with/without parameter adaptation are shown in Figure 4.5 (b). It shows that the two distributions do not have significant difference. Note that, in the past 400 cycles, although 400 measurements are received, there are only 53 events. Therefore, the LS-based diagnosis only runs 53 times. Compared with traditional RS-based diagnosis that needs to run 400 times, the reduction of computation is (400-53)/400=86.8% and computation is 7.55 times faster.

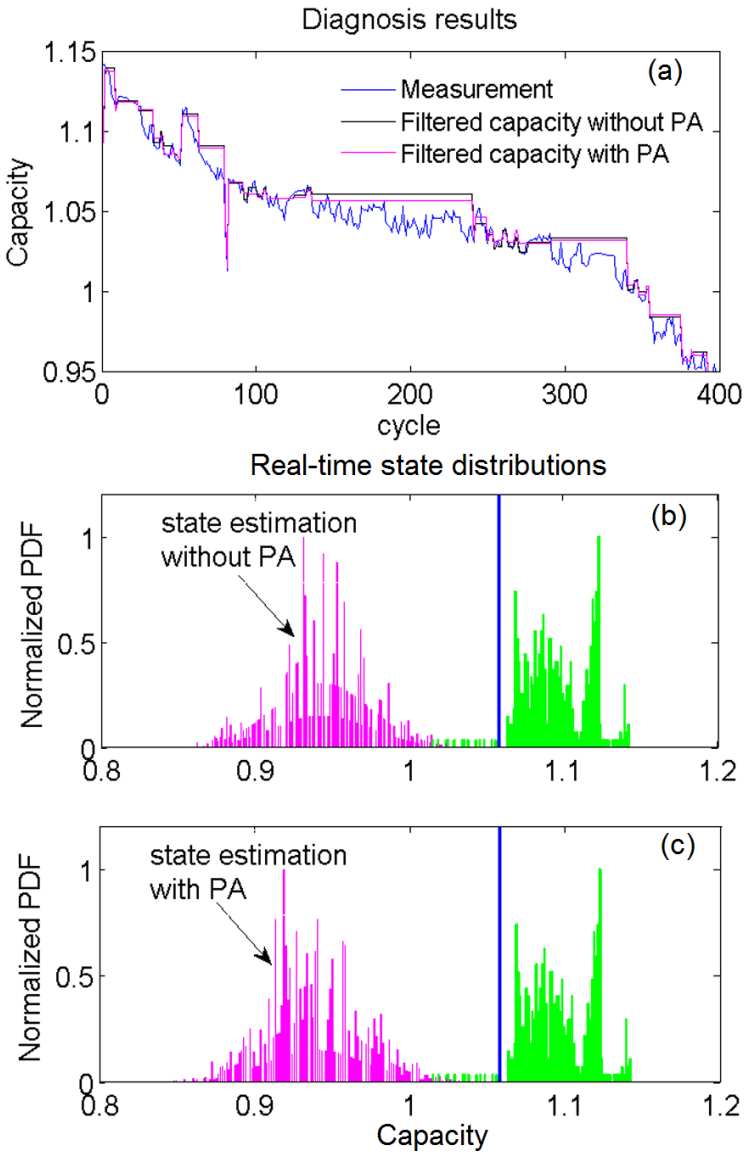


Figure 4.5: The comparison of diagnosis results with/without parameter adaptation. (a): The comparison of capacity from Coulomb counting against the estimated mean value. (b)-(c): The real-time capacity distribution from the algorithm without/with parameter adaptation against baseline.

4.3.2.3 Prognosis results

4.3.2.3.1 Parameter adaptation

Figure 4.6 (a) and (b) compares the prognostic results with 500 particles to approximate the TTF distribution at the 400th cycle.

In this experiment, p in parameter adjustment in Eq. (4.7) is set as 1. Initially, the parameter in the prognosis model is set to be $p_p = 2$. Sub-figure (a) shows the prognosis results without parameter adaptation in the prognosis algorithm. The predicted TTF for this battery is 1102.2 and the RUL is 702.2 cycles. The 95% confidence interval of the TTF is [1021.6 1183.2]. Compared with the ground truth TTF of 810, the difference is 292.4 cycles. This means that if the model parameters are not accurate, the algorithm without parameter adaptation cannot provide an accurate RUL prediction for decision-making.

The prognosis results with parameter adaptation are shown in sub-figure (b) of Figure 4.6. With the proposed method, p_p is adjust to 1.384, the predicted TTF for this battery is 815.2 cycles and the RUL is 415.2 cycles. The 95% confidence interval (CI) of the TTF is [737.5 892.9]. Compared with the ground truth TTF of 810, the difference is 5.2 cycles. The prediction accuracy is significantly improved by the RLS based parameter adaptation approach. The reduction of prediction error is (292.4-5.2)/292.4=98.2%. Note that the 95% CI of predicted TTF in sub-figure (b) is similar and slightly smaller than that in sub-figure (a), which is caused by the reduction of the prediction horizon and uncertainty accumulation.

To illustrate the improvement of prediction accuracy in the entire battery life, the α-λ metrics with α=0.3 for the prognostic results with/without parameter adaptation of the 4 batteries are shown in Figure 4.7. Taking battery CS2-36 as an example, for the result from the algorithm without parameter adaptation, the parameter p_p is set to be 2. Since there is no correction action to adjust p_p, the algorithm is executed with $p_p = 2$ for the whole FDP process. The predicted RUL is longer than the ground truth, which means that the results cannot provide accurate information for CBM and decision-making. On the contrary, the algorithm with parameter adaptation adjusts the initial parameter automatically and achieves much accurate outcomes. At the 79th cycle, parameter adaptation is triggered for the first time, and p_p is adjusted,

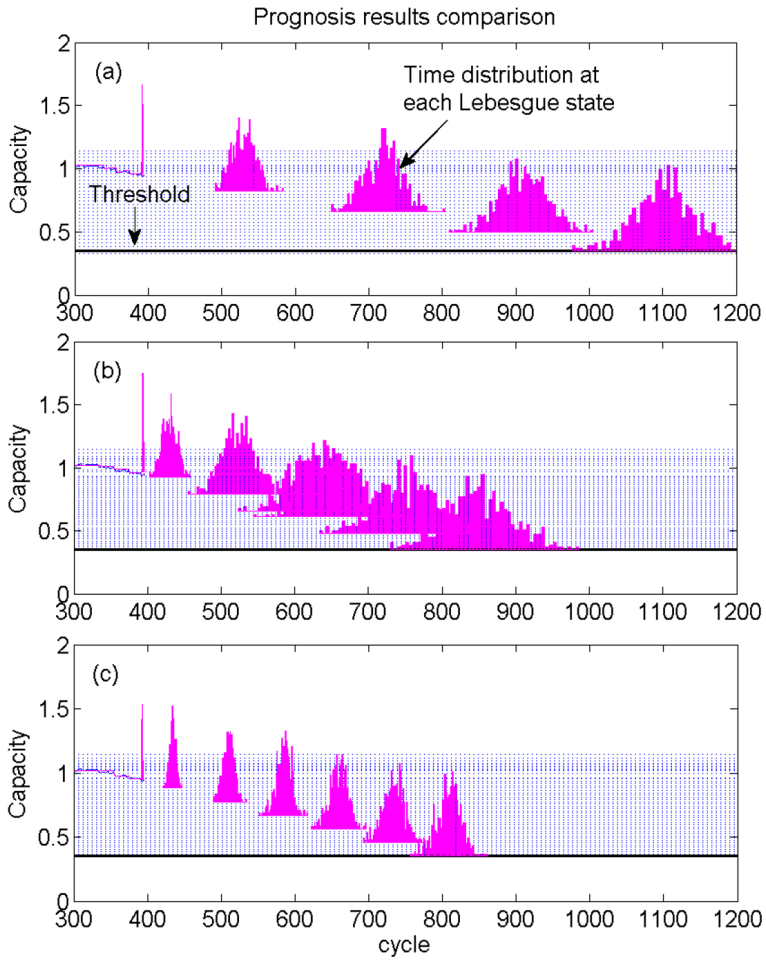


Figure 4.6: LS-based prognosis at the 400th cycle. (a): result without parameter adaptation and noise adjustment, (b): result with parameter adaptation,(c): result with parameter adaptation and noise adjustment

which brings the predicted RUL inside the accuracy zone and stay in the zone in the future prediction.

To further illustrate the parameter adjustment, the parameter adjustments in the diagnosis and prognosis models in the whole FDP process for battery CS2-36 are shown in Figure 4.8. The diagnosis and prognosis processes are executed with the initial value. When an event occurs and the deviation between the priori and posteriori distribution exceeds a predefined threshold, the optimization process is triggered, the diagnosis and prognosis parameters are adjusted. Note that the diagnosis parameter is adjusted based on the state distributions while the prognosis

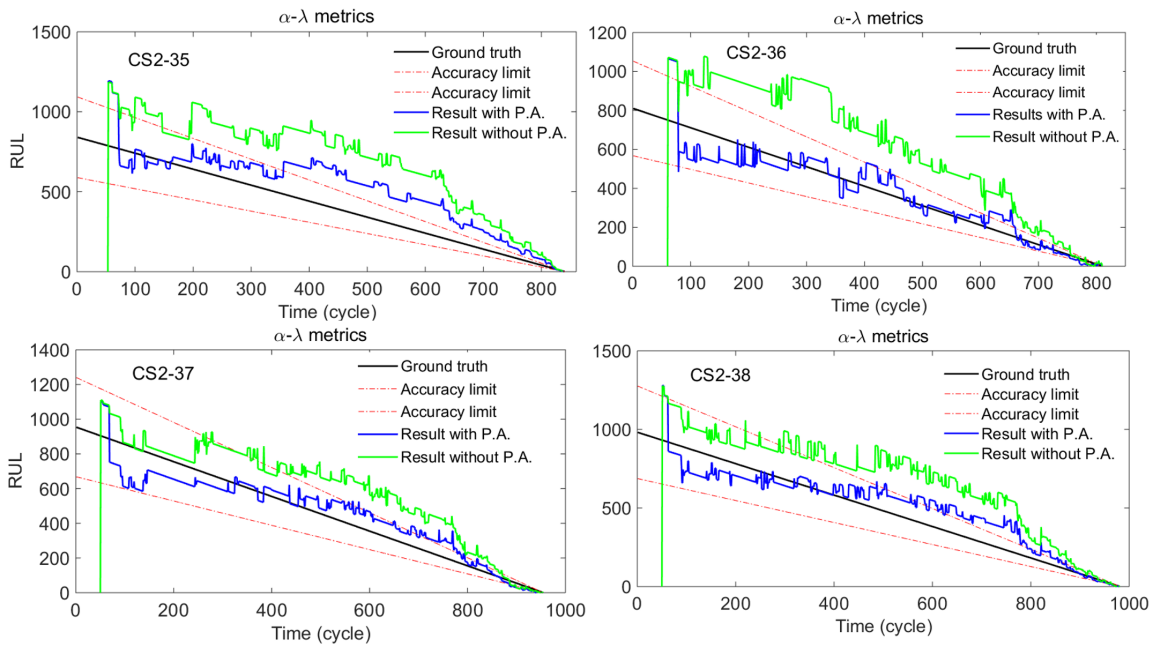


Figure 4.7: Prognostic accuracy comparison for the algorithm with/without parameter adaptation.

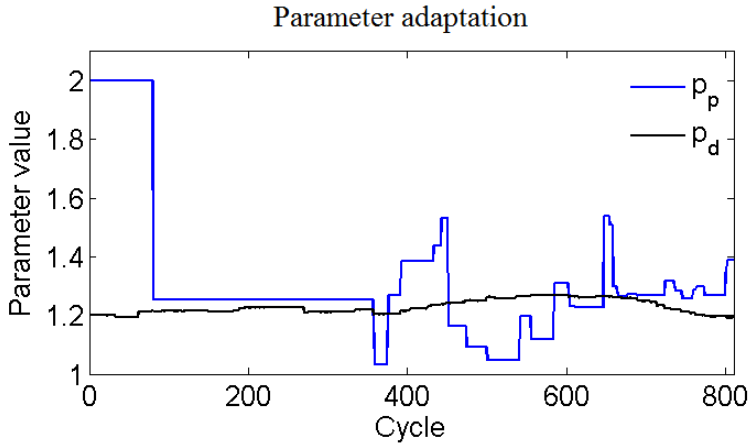


Figure 4.8: Parameter adjustment in LS-FDP of battery CS2-36.

parameter is adjusted according to the time distributions. In Figure 4.8, the flat segments are either no event or event happens but no need to adjust the parameters. The parameter is adjusted only when an event happens and the parameter is adjusted.

4.3.2.3.2 Noise adjustment

In this section, we focus on the noise adjustment in prognosis, which is represented by the noise term in model (4.10). In this experiment, m in noise adjustment Equation (4.8) is set as 1. The initial variance is set to be 50 based on the four battery data set. The uncertainty is accumulated along with the prediction in prognosis, leading to poor TTF distribution prediction if not well managed.

The prognosis is executed with an initial parameter of the noise term. When the noise adjustment is triggered, the variance of the noise term is adjusted.

Figure 4.6 (c) shows the predicted time distributions to reach every Lebesgue state with the noise adjustment and parameter adaptation at the 400th cycle. The variance of pdf at Lebesgue state F_{C+1} from prediction at F_C and diagnosis at F_{C+1}, $\hat{t}_{C+1}$ and t_{C+1}, are 1.7446 and 1.7184, respectively. The variance of the noise term in model (4.10) is adjusted to $\frac{1.7184}{1.7446} = 98.5\%$ of its previous value and the new variance is used in the future prediction steps until the next variance adjustment happens.

The standard deviation of the TTF distribution obtained from the algorithms with/without noise adjustment at the 400th cycle are 16.62 and 38.85, respectively, which means the prediction precision is significantly improved compared with Figure 4.6 (a) and (b), in which the noise terms in the prediction process are not adjusted. The mean of the predicted TTF without and with noise adjustment are 815.2 and 808.4, respectively. Both are close to the ground truth TTF. Note that this is just one example of the noise term adjustment in the whole FDP process, the noise term adjustment will be executed repeatedly as long as the new measurement triggers the adjustment. Judging from the results, we can see that a precise predicted RUL can be obtained by the proposed method even the initial noise variance is large.

In order to show the effect of proposed noise adjustment method during the whole FDP process, the change of the standard deviation of the predicted TTF (σ_{TTF}) with/without noise adjustment are depicted in Figure 4.9. It's obvious that the noise

adjustment reduces the uncertainty in the predicted TTF distribution effectively, and enhances the confidence level for the decision-making and condition-based maintenance. It is worth mentioning that as the battery approaches the end of life, the prediction horizon becomes small and the results with and without noise adjustment converge. This indicates that uncertainty management is critical for prognosis with a large prediction horizon.

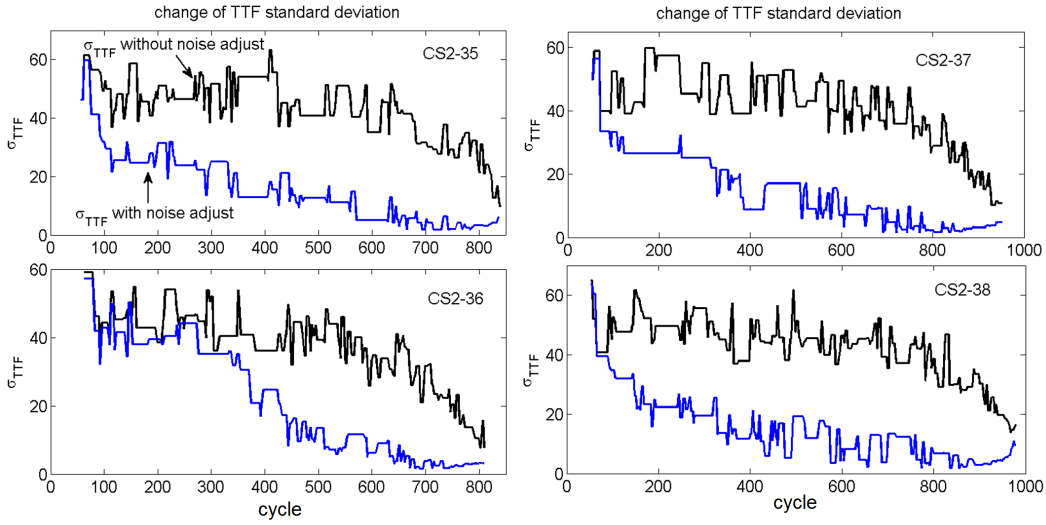


Figure 4.9: The comparison of standard deviation of RUL with/without noise adjustment for 4 batteries.

4.3.2.3.3 Parameter adaption and noise adjustment results analysis

The diagnostic and prognostic results of FDP algorithms with and without the proposed uncertainty management schemes are summarized and compared in Table 4.2. Compared with FDP algorithm without parameter adaptation, the diagnosis result with parameter adaptation shows much more accurate state estimation, the prognosis with parameter adaptation has a much smaller prediction error (292.4 cycles vs. 5.2 cycles). The prognosis results show that the 95% CI of the TTF is reduced from 155.4 to 66.4 cycles by the proposed noise adjustment method, which indicates that the noise adjustment improves the precision of the TTF prediction.

Table 4.2: Comparison of the FDP results at the 400th cycle

	Without PA.	With PA	With PA&NA
μ_C	0.9534	0.9476	0.9447
CI_C	[0.893 1.011]	[0.887 1.005]	[0.889 1.006]
p_d	1.2	1.215	1.199
p_p	2	1.384	1.2
e_P	292.4	5.2	1.6
TTF_P	1102.4	815.2	808.4
CI_t	[1021.6 1183.2]	[737.5 892.9]	[775.2 841.6]

PA: parameter adaptation; NA: Noise adjustment; μ_C: expectation of the estimated capacity; CI_C: 95% confidence interval of the estimated capacity; e_P: prediction error compared with the ground truth; TTF_P: expectation of the predicted TTF; CI_t: 95% confidence interval of predicted TTF.

The above comparison of prediction accuracy is only conducted at the 400th cycle. Two more snapshots at the 500th and 600th cycles are shown in Tables 4.3 and 4.4, respectively. As we can see, the accuracy of the predicted RUL is improved by the proposed parameter adaptation method, the prediction error reduces to 5.2, 70.69, and 39.79 cycles at the 400th, 500th, 600th cycle, respectively. For the precision of the predicted RUL, the 95% CI is reduced by the proposed noise adjustment method to 66.4, 12.5, and 10.8 cycles at 400th, 500th, 600th cycle.

Table 4.3: Comparison of the FDP results at the 500th cycle

	Without PA.	With PA	With PA&NA
μ_C	0.8790	0.9749	0.8640
CI_C	[0.814 0.944]	[0.804 0.946]	[0.799 0.929]
p_d	1.2	1.226	1.1942
p_p	2	1.265	1.2165
e_P	305.9	70.69	63.63
TTF_P	1115.9	880.69	873.63
CI_t	[950.4 1270.1]	[756.19 974.9]	[865.1 877.6]

Table 4.4: Comparison of the FDP results at the 600th cycle

	Without PA.	With PA	With PA&NA
μ_C	0.7860	0.7834	0.7935
CI_C	[0.721 0.851]	[0.709 0.857]	[0.721 0.866]
p_d	1.2	1.1867	1.2131
p_p	2	1.2349	1.2295
e_P	211.57	39.79	44.01
TTF_P	1021.57	849.79	854.01
CI_t	[938.1 1088.2]	[747.3 981.4]	[847.1 857.9]

4.4 Conclusions

In order to enhance the accuracy and precision of the state estimation and predicted TTF or RUL, the model parameters and uncertainties need to be managed. In this Chapter, a recursive least square method is introduced in LS-FDP to adjust the parameters of the diagnosis and prognosis models to accommodate the changing of fault dynamics. A feedback loop based on the differences between the predicted priori time distribution and real-time posteriori time distribution is used to adjust the noise term in the model.

Particle filtering-based algorithms in LS framework with/without parameter adaptation and noise adjustment are developed and verified with an application to the prediction of SOH of Lithium-ion batteries. Experimental results are presented and compared to demonstrate the effectiveness of the proposed parameter adaptation and noise adjustment schemes. The results shows that RUL and TTF from prognosis is significantly improved with parameter adaptation and uncertainty management, which is a strong support for the decision-making in the CBM system.

Chapter 5

Adaptive Lebesgue Sampling-based diagnosis and prognosis

In previous LS-FDP, the Lebesgue length is constant and fixed [91, 92, 98] based on the knowledge of the system. This is not an optimal solution for most nonlinear fault dynamics. Since the Lebesgue states in LS-FDP are selected to ensure the performance for the fastest fault growth scenario, the LS-FDP algorithm can be executed more frequently than necessary when the fault growth is slow. This results in over-provisioning of the real-time system hardware. To accommodate the nonlinearity of fault dynamics, it is desirable to adjust the Lebesgue length adaptively and optimally to meet the demands that the FDP is executed in a low frequency when the fault growth is slow, and in a high frequency when the fault growth is fast.

To achieve this goal, an adaptive Lebesgue sampling-based FDP (ALS-FDP) scheme is proposed in this Chapter. In this scheme, the fault growth speed estimated from diagnosis and prognosis is used in real-time to optimize the adjustment of Lebesgue state length, from which a new set of Lebesgue states is obtained. Compared with the initial Lebesgue state lengths, the new Lebesgue state lengths are increased or decreased, depending on the fault growth speed, and is re-adjusted every time when diagnosis is executed. The prognosis algorithm then predicts the distributions of operating time on these updated Lebesgue states. During the prognosis process, the Lebesgue state lengths for long-term prediction are also adjusted based on the previous and current prediction results.

Assume that the fault growth model can be described by a continuous-time differential Eq. (3.1). In general, the discretized real-time model of this continuous-time system can be described by Eq. (3.2), which needs to be further discussed in adaptive Lebesgue sampling based fault diagnosis and prognosis (ALS-FDP) framework as follows:

5.1.1 LS-based model with adaptive Lebesgue state length

In our previous work in Chapter 3 and Chapter 4, a LSM is developed with uniform Lebesgue state length. Obviously, this selection is not optimal. In practice, the Lebesgue state length must be adjusted optimally and adaptively according to the fault state and fault growth rate at the current event.

With an adaptive Lebesgue state length $D(t_k)$, the LSM of the fault dynamics can be discretized as:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t(D(t_k), \hat{a}(t_k), \dot{\hat{a}}(t_k)) + \omega_a(t_k) \tag{5.1}$$

The model for ALS-based prognosis is given as:

$$t_{k+1} = t_k + g_t(D(t_k), \hat{a}(t_k), \dot{\hat{a}}(t_k)) + \omega_t(t_k) \tag{5.2}$$

Note that the major difference between these models and those in Chapter 4 is that $D(t_k)$ is no longer a constant but online adjusted in realtime. When $D(t_k)$ becomes a constant for all t_k, ALS-FDP reduces to LS-FDP and models (5.1) and (5.2) reduce to Eqs. (4.1) and (4.2), respectively. In addition, to get TTF and RUL distribution on the failure threshold F_f, F_f is always defined as a Lebesgue state.

In our LS-FDP and ALS-FDP algorithm, there is an event checker to determine if the measurement reaches a new Lebesgue state. If the measurements are not

sufficient and suppose that there is a big change of the measurement value, then each measurement triggers an event. In this case, the algorithm will be executed at all the time instants when the measurement is available. This is the worst (most computationally expensive) case, which has the same execution frequency as the RS-FDP. In most cases, LS-FDP and ALS-FDP are more time efficient than RS-FDP.

5.2 ALS-FDP

5.2.1 The framework of ALS-FDP

Figure 5.1 shows the framework of the proposed ALS-FDP. The diagnosis process is started when an event happens with the initial Lebesgue length setting. The Lebesgue length is updated in diagnosis by a gradient-based method according to the fault growth speed. At the time instant when an event happens at state F_d, diagnostic algorithm is executed to estimated the fault state at this time instant and prognostic algorithm calculates the distributions of operating time when the fault reaches Lebesgue states $\{F_{d+1}, ..., F_f\}$. In prognosis, the initial Lebesgue length setting is from the diagnostic process, and it will also be adjusted along with prognosis to accommodate the fault growth nonlinearity.

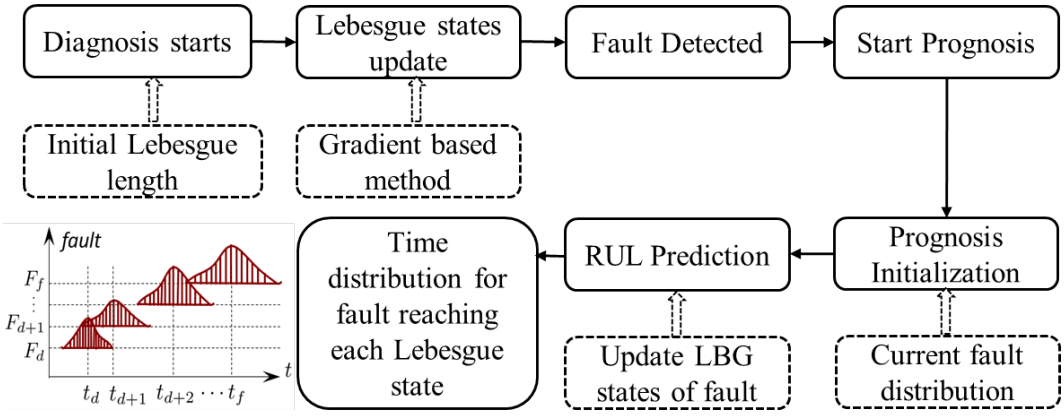


Figure 5.1: Framework of ALS-FDP. (LBG: Lebesgue)

5.2.2 Implementation of ALS-FDP with particle filtering

The details of particle filtering method are shown in Section 3.2.1. In this section, we will only focus on the design of ALS-FDP. In ALS-FDP, the Lebesgue state length is online adjusted in both diagnostic and prognostic processes and is denoted as $D(t_k)$, which is not a constant. This is the main difference from our previous works in Chapters 3 and 4 [91].

5.2.2.1 Adjustment of Lebesgue state length in diagnosis

The adaption scheme of Lebesgue state length, illustrated with a battery degradation data [90], is shown in Figure 5.2.

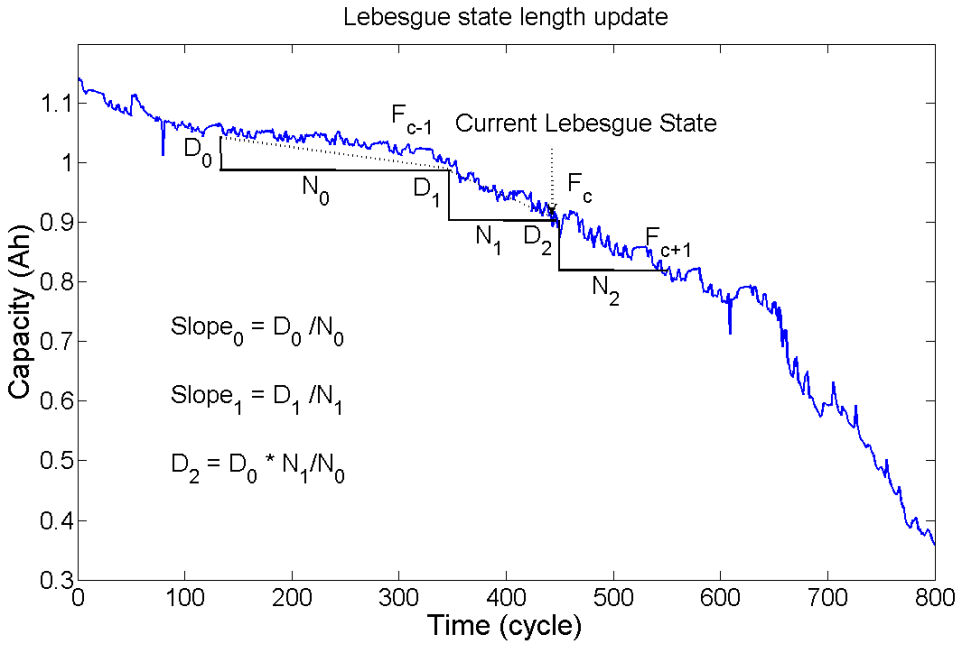


Figure 5.2: Lebesgue state length adjustment in diagnostic process

Initially, diagnostic process is executed with a given initial Lebesgue state length D_0. The slopes given by Lebesgue state lengths, as D_0, D_1, D_2, and the time intervals, as N_0, N_1, N_2, represent the nonlinearity of the fault growth. Here D_0, D_1 are Lebesgue state length in the past and N_0, N_1 are time intervals in the past, while D_2 is

the next Lebesgue state length to be adjusted. The fault growth speed at current and last Lebesgue states, F_c and F_{c-1}, are compared. If the fault growth speed becomes faster, the Lebesgue state length for the next state will be decreased. Otherwise, it will be increased. The Lebesgue state length for the next step is calculated as:

$$D_2 = \frac{D_0/N_0}{D_1/N_1} \cdot D_1 = D_0 \cdot \frac{N_1}{N_0} \tag{5.3}$$

The Lebesgue state length is updated iteratively. It is worth mentioning that this method of Lebesgue state length adjustment has a time delay because the adjustment is based on the past information of D_0, D_1, N_0, and N_1.

5.2.2.2 Adjustment of Lebesgue length in Prognosis

The adjustment of Lebesgue state length in prognosis requires an event has happened and prognostic algorithm is executed. Figure 5.3 illustrates the adjustment of Lebesgue state in prognosis, which is described as follows:

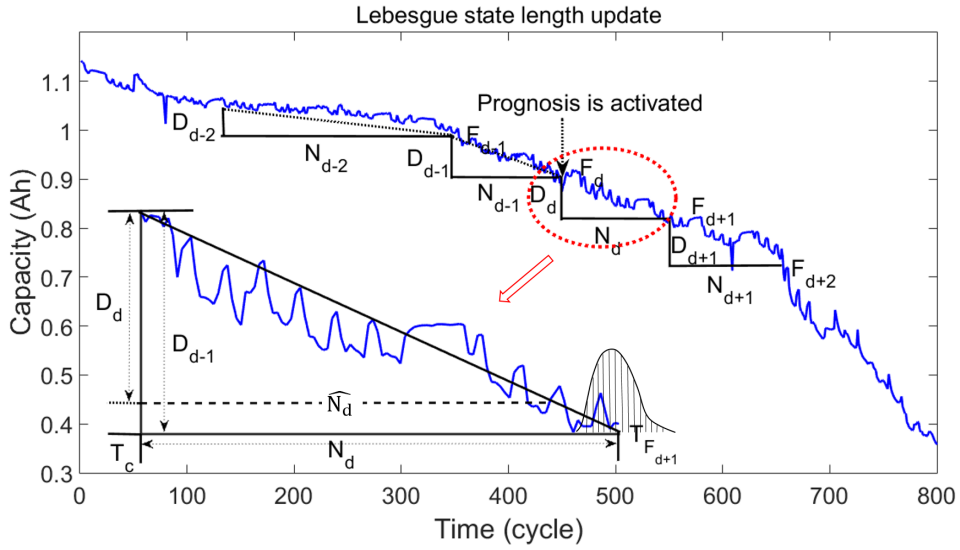


Figure 5.3: The process of Lebesgue state length update during the prognosis process.

1. When the prognostic algorithm is activated at F_d, it is executed with an initial Lebesgue state length D_{d-1} obtained from diagnosis process. The time interval

between Lebesgue state F_{d-1} and F_d is N_{d-1}, the slope is calculated as $S_{d-1} = D_{d-1}/N_{d-1}$;

2. Lebesgue state length for prognosis is D_{d-1} when the prognosis is initialized on Lebesgue state F_d. The time to reach F_{d+1} is given by (4.2) $N_d = T_{F_{d+1}} - T_c$ for the prognosis on Lebesgue state F_d, where $T_{F_{d+1}}$ is the mean of the time distribution for fault state reaching F_{d+1}, T_c is the current time instant. The Lebesgue state length between Lebesgue state F_d and F_{d+1} is updated by $D_d = S_{d-1} \times N_d$, the new slope is $S_d = D_{d-1}/N_d$;

3. The prognosis at Lebesgue state F_{d+1} starts with Lebesgue state length D_d and slope S_d, time interval N_{d+1} is also calculated by (4.2). Lebesgue state length D_{d+1} is given as $D_{d+1} = S_d \times N_{d+1}$, then calculate the slope $S_{d+1} = D_d/N_{d+1}$;

4. Repeat steps 2) and 3) until the Lebesgue state reaches the failure threshold F_f. This step yields the TTF and RUL distributions;

5. At the next time instant with a new event happens, the time interval is updated by the ground truth of cycle life to $\bar{N}_d$, the slope $\bar{S}_d = D_{d-1}/\bar{N}_d$ and D_d are used as the initial condition for the new prognostic loop. Note that D_d is adjusted in diagnostic process. The process for the prognosis repeats step 1) to 4) with updated D_d as the initial value.

Note that, during the prognostic process, the Lebesgue state length is changed after one step forward prognosis. The adjusted Lebesgue states are used for the following prognostic process from F_{d+1} to F_{d+2} as the initial set of Lebesgue states. After the entire prognosis from F_d to F_f, the whole set of Lebesgue states gets adjusted and becomes the initial condition for the next diagnosis and prognosis process. By adjusting the Lebesgue state length this way, FDP will be executed with different frequencies, determined by the fault growth speed at different stages.

Remark: The noise in the data will not affect the adaptive algorithm. It is known that Bayesian estimation approach is able to handle noises. For linear systems with Gaussian noises, Kalman filter is able to provide an optimal solution. However, since battery capacity degradation is a nonlinear process with non-Gaussian noise, particle filtering algorithm is used in this Chapter. To deal with uncertainties from various resources, noise terms are added in models (5.1) and (5.2), which are not necessary to be Gaussian noises.

5.3 Experimental results

In this section, the proposed ALS-FDP scheme based on a particle filtering algorithm is demonstrated with an application to the capacity degradation of a set of Li-ion batteries, the results are compared against those from RS-FDP and LS-FDP to illustrate the advantages of ALS-FPD. Battery is a safety critical component that provides power for most autonomous systems, such as computers, robots, electrical vehicles, and unmanned aircraft [4, 18, 19]. Since the life and state of the batteries are not directly observable, diagnosis and prognosis are critical for estimating the battery state [15, 16, 81, 99], such as state-of-health (SOH), state-of-charge (SOC), and remaining useful life (RUL).

The experiment investigates the SOH of four identical Li-ion batteries with 1.1 Ah rated capacity. The degradation of the capacities is descried in Section 3.3. The failure threshold for the SOH is set to be 0.35 Ah and the battery capacity reaches this threshold at the 840th, 810th, 954th, and 981st cycle, respectively. From this series of experimental data, an empirical model will be established and will be used in diagnosis and prognosis, as discussed below.

Since measurements of the battery capacity are noisy, the measurement values show fluctuation. The fluctuation happens around the Lebesgue state will trigger the execution of the algorithms. If all the fluctuations happen around the Lebesgue

states, this could lead to the situation with the most executions and the number of executions of LS-FDP is the same as that of RS-FDP. Otherwise, the noise in the measurements will not result in too many executions of the algorithm. In this research, with noisy capacity measurement data from 4 batteries, our algorithm shows significant reduction in computation.

5.3.1 LS-FDP

In LS-FDP, the feature value range is divided into a series of Lebesgue states. If a new measurement causes a transition of Lebesgue state, *i.e.*, an event happens, the diagnostic algorithm is executed. The sequence of the event stamps formulates a time series that is used as the input of real-time diagnostic algorithms. The output of fault diagnosis is the fault state distributions at the current time instant, which is compared against the baseline pdf from healthy system to calculate the probability of fault.

To implement LS-FDP for the battery capacity degradation, 40 uniformly distributed Lebesgue states are defined in the battery's full capacity of 1.1 Ah. With this setting, the diagnostic algorithm is executed only when the capacity degrades from one Lebesgue state to another. The model for diagnosis is given in Eq. (4.9).

Figure 5.4 shows the diagnostic results at the 472nd cycle. The particle filtering algorithm uses 500 particles. The expected value of battery capacity is 0.885 and the 95% confidence interval is [0.828, 0.947]. Same as the example in RS-FPD, the upper sub-figure shows the comparison of capacity from Coulomb counting (blue) against the estimated mean value from diagnosis (magenta). Note that the flat magenta segments mean no event and diagnosis is not executed. The lower sub-figure shows the real-time capacity distribution at the current cycle against the baseline. In the past 472 cycles, although 472 measurements are received, there are only 76 events. Therefore, the LS-based diagnosis only runs 76 times. Compared with traditional

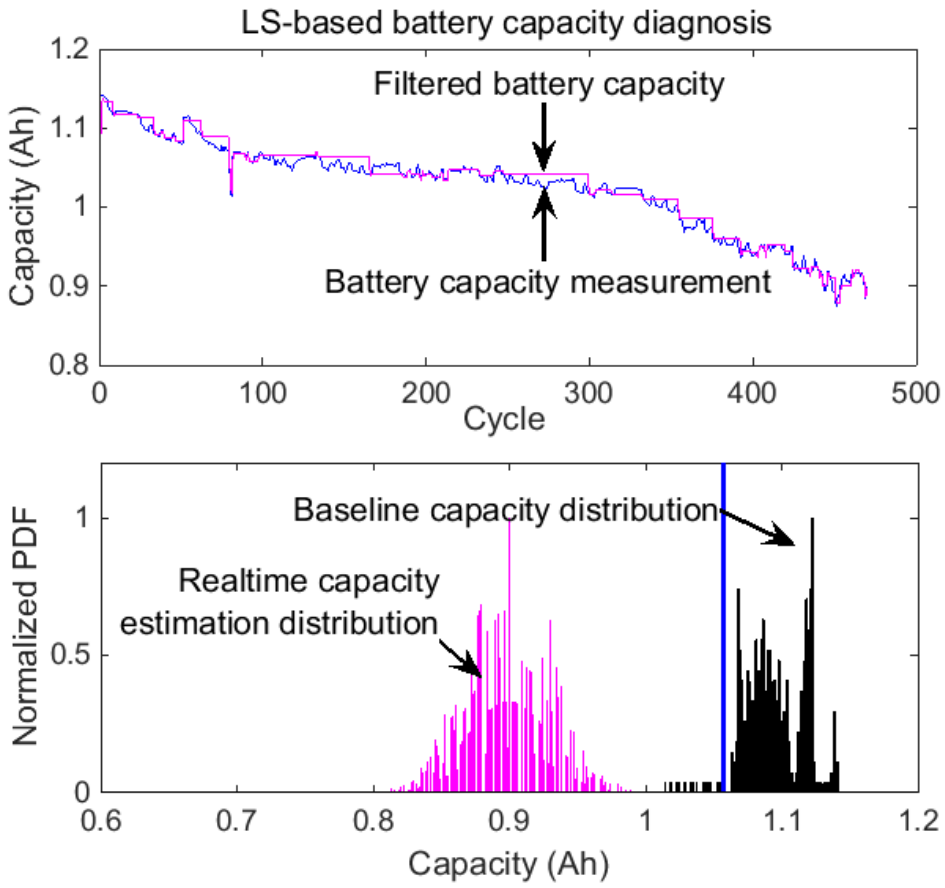


Figure 5.4: LS-based diagnosis for battery at the 472nd cycle.

RS-based diagnosis that needs to run 472 times, the reduction of computation is (472-76)/472=83.9% and computation is 6.21 times faster in running numbers.

Different from the diagnosis that yields fault state distribution at each time instant when an event occurs, prognosis estimates the time distribution of fault state reaching each Lebesgue state. The output of diagnosis is a capacity distribution at current time instant. It cannot be used for prognosis directly and has to be transformed into the operation time distribution. To implement prognosis in LS-FDP framework, the operation time distribution is achieved as discussed in Section 3.1.4.

LS-based prognosis is conducted on fault dimension axis to predict the time-to-Lebesgue-state directly. The model for prognosis is given as Eq. (4.10) [92].

Figure 5.5 shows the prognostic results with 500 particles at the 472nd cycle.

The predicted TTF for this battery is 770.5 and the RUL is 298.5 cycles. The 95% confidence interval of the TTF is [740.4 786.7]. Compared with the ground truth TTF of 810, the difference is 39.5 cycles. The LS-based prognosis only has a prognostic horizon of 23 Lebesgue states and can afford 500 particles. The reduction of computation time is (2.309-0.006)/2.309=99.74% and the computation is about 384 times faster. Note that in RS-based prognosis, only 20 particles (at the cost of performance) are used to make real-time implementation possible. Note also that LS-FDP offers better performance than RS-FDP in terms of TTF prediction due to a short prognostic horizon.

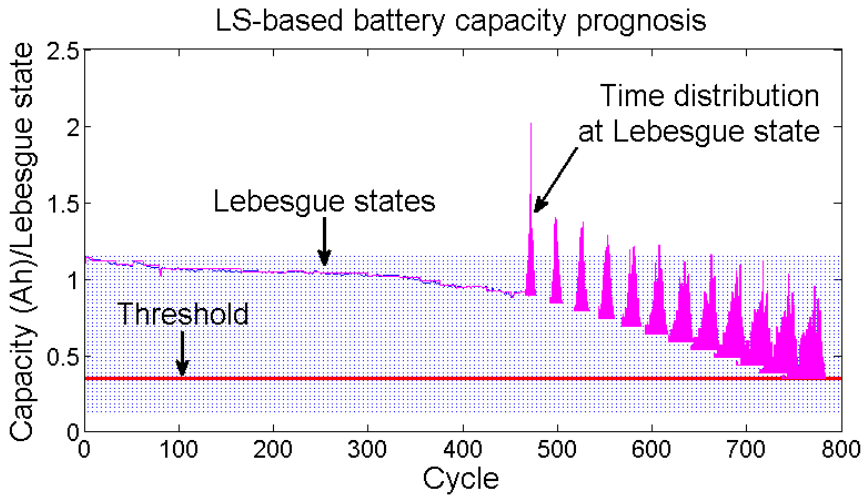


Figure 5.5: LS-based prognosis at the 472nd cycle.

5.3.2 ALS-FDP

In ALS-FDP, the battery's full capacity range is initially divided into 40 Lebesgue states as in LS-FDP. In the FDP process, Lebesgue state lengths are adjusted to optimize the usage of computation resources.

The diagnostic model of ALS-FDP is given as:

$$C(t_{k+1}) = C(t_k) - p_d \cdot C(t_k) \cdot D(t_k) \cdot sgn(C(t_k) - C(t_{k-1})) + \omega_C(t_k) \qquad (5.4)$$

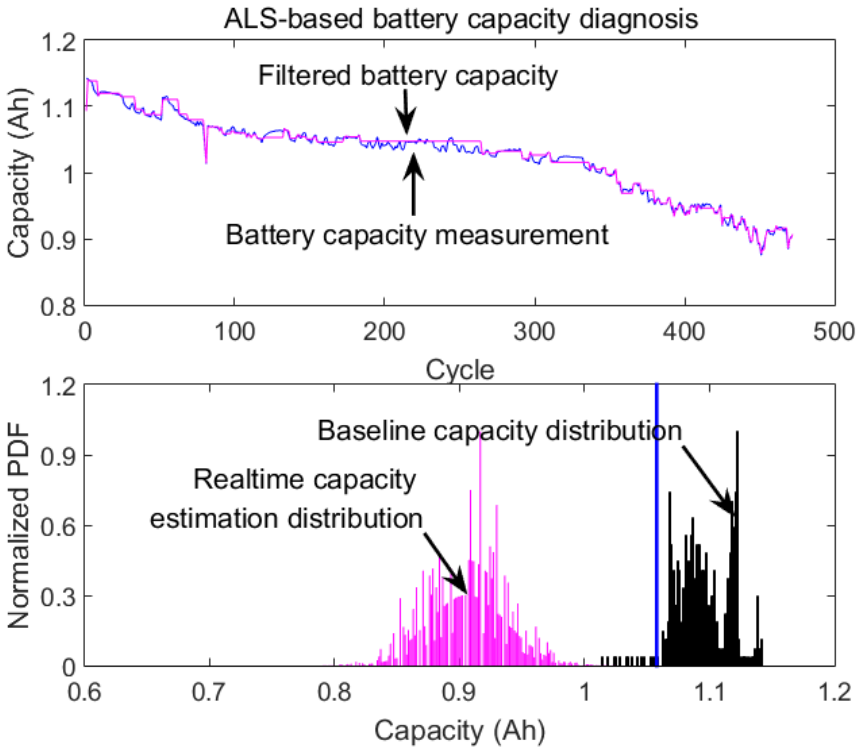


Figure 5.6: ALS-based diagnosis for battery at the 472nd cycle.

the difference between (5.4) and (4.10) is that the Lebesgue length for every Lebesgue state is no longer uniform.

Note that since Lebesgue length in ALS-FDP is not a constant, $D(t_k)$ is used, which indicates that it is adjusted online. The particle filtering algorithm uses 500 particles. Figure 5.6 shows the diagnostic results at the 472nd cycle. The upper sub-figure is the battery capacity measurement by Coulomb counting method (blue) compared with the filtered capacity estimation (magenta). The diagnosis process is executed 72 times, which is a little smaller than the LS-FDP. The reason is that at early stage of the degradation, the degradation speed is low, and the Lebesgue state length for the previous diagnosis is increased, which results in a reduced number of diagnosis execution. The lower sub-figure shows the comparison of battery capacity distribution with baseline distribution. The expected value of battery capacity is 0.8887 and the 95% confidence interval is $[0.8287,\ 0.9505]$.

The ALS-FDP prognosis is conducted with 500 particles to predict the time-to-Lebesgue-state, the prognostic model is given as:

$$t_{k+1} = t_k + p_p \cdot C(t_k) \cdot D(t_k) \cdot exp\left(-\dot{C}(t_k)\right) + \omega_k(t_k) \tag{5.5}$$

Again, $D(t_k)$ indicates that Lebesgue length is online adjusted in prognosis.

Figure 5.7 shows the prognostic result at the 472nd cycle. In this figure, the prediction horizon is 31, which is slightly larger that of LS-FDP with uniform Lebesgue length. To make the figure clear, only the time distribution at selected Lebesgue states are plotted. The predicted TTF for this battery is 774.1 and the RUL is 302.1 cycles. The 95% confidence interval of the TTF is [736.5 790.3]. Compared with the ground truth TTF of 810, the difference is 35.9 cycles. Note that Lebesgue states for ALS-based prognosis are distributed unequally along the state axis, which is different from the case of LS-based prognosis.

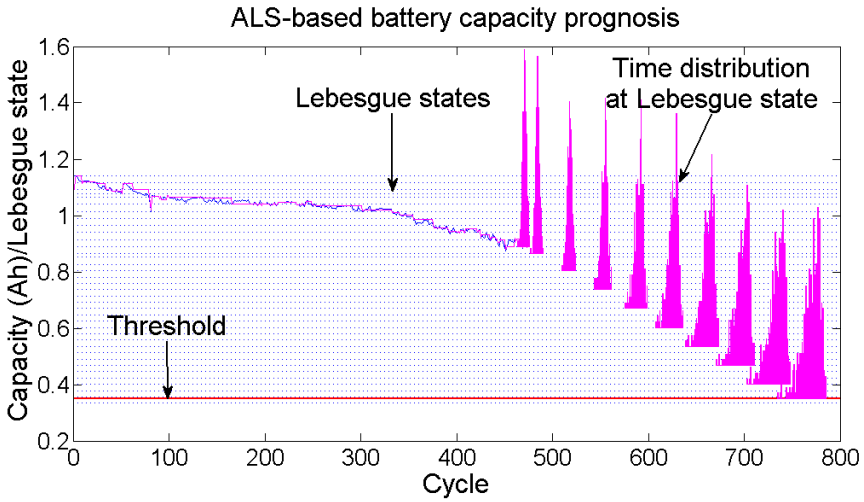


Figure 5.7: ALS-based prognosis at the 472nd cycle.

Note that the prognostic model is not a function of time. The noise of the prognostic model is defined in the Lebesgue sampling framework. With a constant Lebesgue length, the noise term remains the same. In this Chapter, the Lebesgue length is adjusted according to the fault dynamics. The adjustment of Lebesgue state length will cause change of noise, which is not considered in this research because the

uncertainty and the Lebesgue state length are both bounded. Experimental results show that change of Lebesgue state length has trivial influence on noise. However, the change of noise caused by the adjustment of Lebesgue length needs to be studied in a simulation environment and will be one of our future works.

5.3.3 Comparison of RS-FDP, LS-FDP, and ALS-FDP

The performance of RS-FDP, LS-FDP, and ALS-FDP, are summarized in Table 5.1. It is clear that the three methods have comparable performances for diagnosis. LS-FDP and ALS-FDP are both time efficient than RS-FDP in diagnostic prospective.

In terms of prognosis, LS-FDP and ALS-FDP show better performances in different aspects. First, LS-FDP and ALS-FDP have better accuracy and precision than RS-FDP by comparing the confidence interval (CI) and means of the TTF distributions. Second, LS-FDP and ALS-FDP avoid the large amount of uncertainty accumulation during the long-horizon prediction, which is demonstrated by the smaller CI compared with the RS-FDP. More importantly, ALS-FDP dynamically distributes the computation resources according to fault growth speed. That is, when the fault growth is slow, FDP will execute less frequently; otherwise it executes more frequently. Compared with RS-based prognosis with a horizon of 758 cycles and small number (20) of particles at the 472th cycle, the LS-based and ALS-based prognosis have a horizon of 23 and 31 Lebesgue states, respectively, and can afford 500 particles. The computation time for LS-based and ALS-based prognosis are only 0.26% and 0.29% of that of the RS-based prognosis, respectively. Compared to LS-based prognosis, ALS-based prognosis has a little more but comparable computational time. The reason is that ALS-based prognosis reduces the Lebesgue state length and has more Lebesgue states after adjustment for better long-term prediction.

Above results only show the comparison at the 472nd cycle of one battery case. To compare the performance in the whole battery degradation process, FDP accuracies

Table 5.1: Comparison of RS-FDP, LS-FDP, and ALS-FDP

	RS-FDP	LS-FDP	ALS-FDP
Diag. particles	500	500	500
Capacity mean	0.8808	0.8853	0.8887
Capacity 95% CI	[0.852 0.907]	[0.828 0.946]	[0.829 0.951]
Execution No.	472 (100%)	76 (16.1%)	72 (15.3%)
Prog. particles	20	500	500
True TTF	810	810	810
TTF Mean	735	770.5	774.1
TTF 95% CI	[603 1230]	[740.4 786.7]	[736.5 790.3]
Prog. horizon	758	23	31
Computing time (s)	2.309 (100%)	0.0060 (0.26%)	0.0067 (0.29%)

of all the four batteries are studied and compared in terms of $\alpha - \lambda$ metrics [83].

Figure 5.8 shows the α-λ metrics with α=0.3 for four sets of battery capacity degradation data. Because of the short prediction horizon and small uncertainty accumulation, the prediction accuracy for LS-FDP and ALS-FDP are higher than that of RS-FDP. The mean RUL of LS-FDP and ALS-FDP reach the accuracy zone quickly, and the prediction results are stable. The result of RS-FDP exceeds the accuracy limits more often, which means the estimation of RUL is not uniformly accurate, and the 95% CI is much bigger than that of LS-FDP and ALS-FDP, as shown in Table 5.1. The high prediction accuracy of LS-FDP and ALS-FDP is achieved with much lower computation cost compared with RS-FDP.

The application of Lebesgue sampling in FDP provides a natural solution for real-time FDP implementation, especially for those systems with limited computation resources. The ALS-FDP reduces the requirements on computational resources dynamically within a system. Similar to LS-FDP, the prediction horizon of ALS-FDP is very small compared with that of RS-FDP, which is beneficial for managing the uncertainty in prognosis.

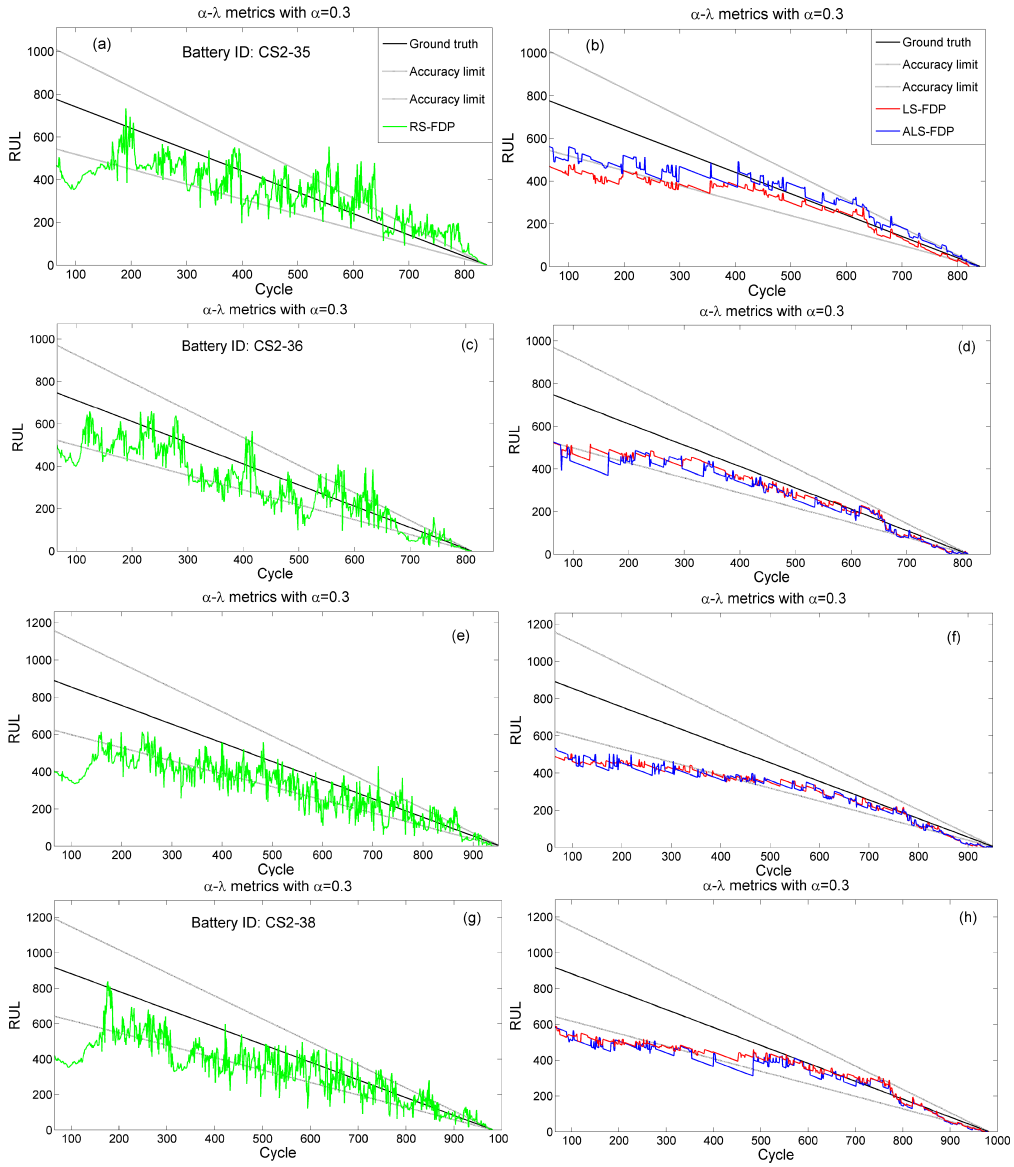


Figure 5.8: Prognostic accuracy of RS-FDP (green), LS-FDP (red), and ALS-FDP (blue).

5.4 Conclusions

In previous chapters, a new LS-FDP is introduced with a philosophy of "execution when needed" to reduce the computation, which makes long-term prognosis possible to be implemented and deployed on embedded systems and micro-controllers. Based on that, this Chapter proposes an ALS-FDP, in which Lebesgue state length is online

adjusted to accommodate the nonlinear fault dynamics. With this new feature, the FDP algorithms are executed more frequently when the fault growth is fast and less frequently when it is slow. This optimizes the usage of system computation resource, makes the computation less expensive, and reduces the cost of algorithm calculation. In ALS-FDP, models for diagnosis and prognosis are designed separately because diagnosis estimates the fault state while prognosis predicts the time for fault state reaching each Lebesgue state directly. A particle filtering-based algorithm is developed with an application to the SOH of Li-ion batteries. Experimental results for RS-FDP, LS-FDP, and ALS-FDP are presented and compared to demonstrate the effectiveness of the proposed ALS-FDP scheme.

Chapter 6

LS-EKF algorithms for SOH and SOC estimation

Battery is a safety critical component to many autonomous systems. It is important to improve the performance of the battery management system (BMS) to make battery operation and maintenance safe, reliable, and cost-efficient. BMS is designed to evaluate the instantaneous state-of-charge in the battery and at the same time to monitor the slowly varying state-of-health. It's essential to accurately estimate the SOC and SOH in order to maximize and optimize the operation of the system. Lastly, knowing the SOC also helps prevent overcharge and over discharge of batteries, which is vital for safe operation and long cycle life of lithium-ion batteries. However, both SOC and SOH are not directly observable, which requires estimation and prediction algorithms, such as diagnosis and prognosis, to provide real-time battery health states [15, 81, 99].

SOC is an indicator that represents the available charge stored in the battery compared to the full capacity charge of the battery, given by a percentage of the current charge to the entire charge [100]. An accurate estimation of SOC is necessary not only for optimal management of the energy stored in the battery, but also to protect the battery from being over-discharged or overcharged.

SOH describes the physical condition of a battery. Unlike SOC, there is no widely accepted definition of SOH. A general definition of SOH is that it quantifies the battery's ability to store energy and deliver specified performance compared to a

fresh battery [20]. The definition of SOC and SOH can be illustrated by Figure 6.1, the rated capacity is the capacity of a fresh battery, the inactive capacity increases with the increasing of the battery service time, resulting in a reduce of SOH. The ratio between the stored energy (green) and total active capacity is the SOC.

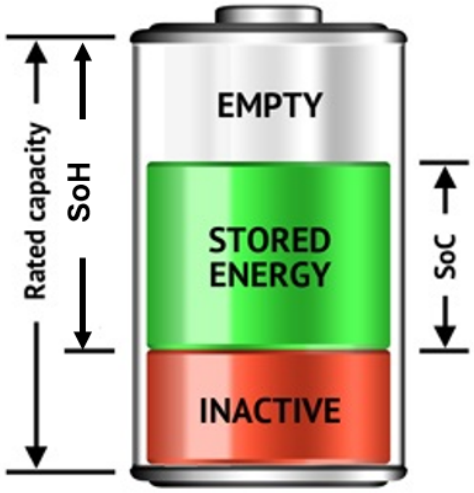


Figure 6.1: Illustration of the SOH and SOC definitions [101]

Despite the importance, the SOC cannot be measured directly from the battery terminals. The easiest way is based on current integration [102–106]. The total capacity of the battery is evaluated in terms of Amp-hours (Ah), which basically is a dimension of electric charge. SOC estimation methods are also developed using reduced state-space electrochemical models, artificial neural networks, and impedance spectroscopy [50, 78, 107, 108]. These methods usually require a large computational cost and very accurate measurements, which make them unsuitable for SOC online application since the estimation of SOC requires a much faster and more accurate algorithms [100, 109], whereas SOH estimation is not so tough.

Extended Kalman filter (EKF) is widely used to estimate the state of a nonlinear state-space system with Gassian noise due to the simplicity [110, 111]. EKF linearizes the nonlinear system model, and calculates the mean and variance of the system state based on the linearized system model and Bayesian theory. Traditionally, EKF is

designed in Riemann sampling framework (RS-EKF), in which the samples are taken in a periodic way (also called "Riemann sampling (RS)"). LS-based fault diagnosis and prognosis (LS-FDP) algorithms are developed with the advantage of reducing the computation requirement since the algorithms are executed "only when necessary" [92].

As discussed in previous chapters, LS-FDP requires two different models for diagnosis and prognosis. The diagnosis model describes the fault growth mechanism. Prognosis model describes the operation time to reach each predefined fault state, the prediction horizon is in the fault dimension axis and described by the number of Lebesgue state, which is relatively short and results in less computation requirement and less uncertainty accumulation.

In this Chapter, a LS-based EKF (LS-EKF) algorithm is designed to estimate SOC and SOH and predict RULs in terms of SOC (battery runs out of charge) and SOH (battery capacity degrades to a threshold measured by charge-discharge cycles), which takes advantage of the merits of EKF and LS. The SOH is firstly estimated and the remaining useful life is predicted by RS-EKF and LS-EKF. Then, the estimated SOH is used as the initial battery capacity to produce the SOC estimation and prediction. By this means, the SOH and SOC estimation and prediction are calculated repeatedly and interconnected during the whole service of the batteries. The results obtained from LS-EKF are compared against those from RS-EKF, which show that LS-EKF has better performances with less computation requirements.

6.0.1 Scheme of SOH and SOC estimation

Since SOC estimation is influenced by SOH, the parameters and initial condition for SOC need to be updated according to SOH. In this Chapter, the battery capacity is estimated by LS-EKF, which is used as the initial condition for SOC estimation. An ECM model-based method is employed to calculate OCV-SOC curve, which is used for

battery SOC diagnosis. The integration of SOC, SOH, and ECM-model are illustrated in Figure 6.2. In this figure, $i(t)$ and $V(t)$ are the measured current and voltage. The OCV-SOC diagnosis and prognosis is executed during each charge-discharge cycle with the SOH estimation from previous $(k-1)$-th cycle as the initial condition to update the total capacity Q_0 in Eq. (6.9). The SOC is acquired by Coulomb counting method and the OCV is calculated by ECM model shown in Figure 6.4. SOC prognosis is conducted based on the prognosis model, which will be discussed later. When the OCV-SOC estimation is completed for the k-th cycle, the collected SOH data by Coulomb counting method is used as the input of the SOH diagnosis and prognosis at the k-th cycle. With the SOH estimation as the initial condition, the SOH prognosis is conducted, and a RUL pdf is achieved. The procedure of the SOH prognosis is discussed in the previous chapters.

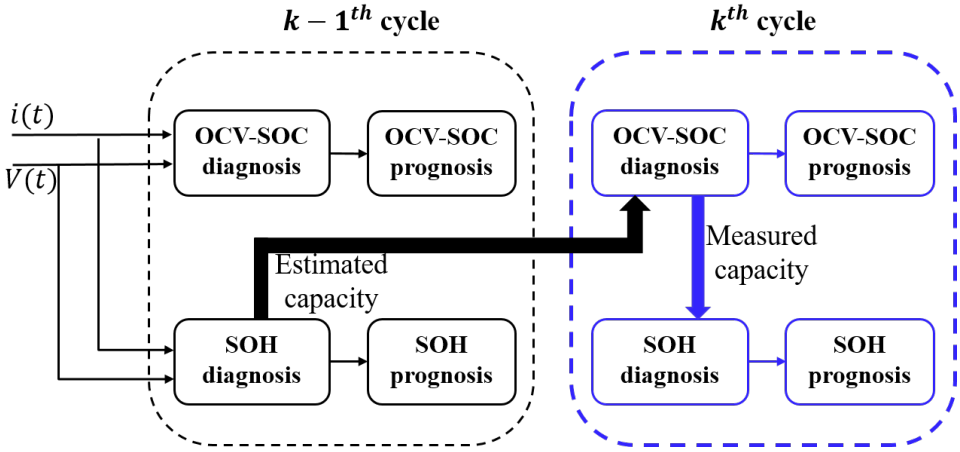


Figure 6.2: Scheme of the proposed SOC and SOH estimation algorithms

Same as the execution of LS-EKF based SOH estimation algorithm, the SOC estimation is conducted if a new measurement reaches the predefined Lebesgue states with initial condition of battery capacity. The remaining service time during one charge-discharge cycle is predicted by the prognosis part of SOC estimation algorithm. When the SOC estimation is completed, a new capacity measurement becomes available and is used to trigger the SOH estimation and RUL prediction.

The achieved capacity is then used to update the initial battery capacity for the SOC estimation in the next cycle. Iteratively, the SOC and SOH are estimated during the service life of the battery by this means.

6.1 The proposed LS-EKF method

6.1.1 LS-EKF for SOH diagnosis

EKF is a recursive algorithm that estimates the true state of a nonlinear system based on noisy measurements. It assumes that the nonlinear system dynamic model is linearized using Taylor expansion at a local point and the noise is subject to Gaussian distribution. EKF has been used for state estimation and RUL predcition of electronic components and battery management [51, 79, 111] due to its low computational requirement.

Traditional EKF algorithms are developed based on RS framework with the advantages of easiness in system analysis and algorithm design. However, it is unfavorable from the computation-efficiency point of view in some cases, especially when the battery SOH degradation is slow. The algorithms are executed as long as a new measurement is available no matter how much new information is included in the measurement. Ideally, it is expected to reduce the executions of algorithm when battery SOH degradation is slow and increase the number of algorithms execution when battery SOH degradation becomes fast. To achieve this goal, Lebesgue sampling method is developed [91, 92, 98]. In this section, the design of EKF in LS framework will be discussed with an example of diagnosis of battery SOH degradation.

In LS-EKF, the general form of SOH diagnosis model is given as [92]:

$$\hat{x}_{t_{k+1}} = f(\hat{x}_{t_k}, u_{t_k}, D_{t_k}) + \omega_{t_k} \tag{6.1}$$

where f is a nonlinear function, u_{t_k} is the input, $\hat{x}_{t_k}$ is the SOH state, ω_{t_k} is the Gaussian noise with covariance Q_{t_k}, which models the uncertainties, $D_{t_k} =$

$L_{t_k,j+1} - L_{t_k,j}$ is the Lebesgue state length given by the distance between two adjacent Lebesgue states. Note that the Lebesgue states $L_{t_k,j}$ are adjusted according the battery degradation speed and, therefore, D_{t_k} is a nonlinear term. Here subscript t_k is the event stamp [92].

The observation model that describes the relationship between state x_{t_k} and measurements z_{t_k} is given by:

$$z_{t_k} = h(x_{t_k}) + v_{t_k} \tag{6.2}$$

where z_{t_k} is the measurement, $h(\cdot)$ is the measurement function, which can be linear or nonlinear, v_{t_k} is a zero-mean Gaussian noises with covariance matrix R_{t_k}. The measurement is used to compare with the predefined Lebesgue state $L_{t_k,i}$. If the measurement remains in the same Lebesgue state, the EKF algorithm will not be executed. Whenever the measurement reaches a new Lebesgue state, it triggers an event and the LS-EKF will be executed as follows.

The Jacobian of $f(\cdot)$ and $h(\cdot)$ are given by:

$$F_{t_{k-1}} = \frac{\partial f}{\partial x} \mid_{\hat{x}_{t_{k-1}|t_{k-1}}} \qquad H_k = \frac{\partial h}{\partial x} \mid_{\hat{x}_{t_k|t_{k-1}}} \tag{6.3}$$

Like other Bayesian methods, EKF algorithm includes two steps: the first step (prediction) is to propagate the state vector x into the next time step by using the state transition model; the second step (update) is to correct the prediction from the first step by using the measurement z. The prediction step can be described as:

$$\begin{aligned} \hat{x}_{t_k|t_{k-1}} &= f(\hat{x}_{t_{k-1}|t_{k-1}}, u_{t_{k-1}}, D_{t_{k-1}}) \\ \Gamma_{t_k|t_{k-1}} &= F_{t_{k-1}} \Gamma_{t_{k-1}|t_{k-1}} F_{t_{k-1}}^T + Q_{t_{k-1}} \end{aligned} \tag{6.4}$$

where $\Gamma_{t_k|t_{k-1}}$ and $Q_{t_{k-1}}$ are the covariance matrices of the predicted state and the process noises.

The equations for the update step are expressed as:

$$
\begin{aligned}
K_{t_k} &= \Gamma_{t_k|t_{k-1}} H_{t_k}^T (H_{t_k} \Gamma_{t_k|t_{k-1}} H_{t_k}^T + R_{t_k})^{-1} \\
\Gamma_{t_k|t_k} &= (I - K_{t_k} H_{t_k}) \Gamma_{t_k|t_{k-1}} \\
\hat{x}_{t_k|t_k} &= \hat{x}_{t_k|t_{k-1}} + K_{t_k}(z_{t_k} - h(\hat{x}_{t_k|t_{k-1}}))
\end{aligned}
\tag{6.5}
$$

where K_{t_k} is the near-optimal Kalman gain, R_{t_k} is the covariance matrix of the observation noises, $\Gamma_{t_k|t_k}$ is the updated covariance estimate, $\hat{x}_{t_k|t_k}$ is the updated state estimate, and I is the identity matrix. Note that the Jacobian needs to be calculated with the predicted state at each instant.

6.1.2 LS-EKF for SOH prognosis

As described in Chapter 3, the LS-EKF based prognosis predicts the operation time reaching each predefined Lebesgue state and this involves a model that describes the battery operation time as a function of battery capacity degradation. With this understanding, the prognosis model is given as:

$$
\hat{t}_{k+1} = g(\hat{t}_k, \hat{x}_{t_k}, D_{t_k}) + \tau_{t_k} \tag{6.6}
$$

where $\hat{t}_k$ is the operation time distribution when the system state reaches the k-th Lebesgue state L_k, g is a nonlinear function, τ_{t_k} is a Gaussian noise term with covariance S_{t_k}.

Since prognosis involves long-term prediction without new measurement, there is no update step in the prognosis. Therefore, the prognosis will only conduct the prediction given as:

$$
\begin{aligned}
\hat{t}_{k|k-1} &= g_t(\hat{t}_{k-1|k-1}, \hat{x}_{t_{k-1}|t_{k-1}}, D_{t_k}) \\
\Upsilon_{t_k|t_{k-1}} &= G_{t_{k-1}} \Upsilon_{t_{k-1}|t_{k-1}} G_{t_{k-1}}^T + S_{t_{k-1}}
\end{aligned}
\tag{6.7}
$$

where $G_{t_{k-1}} = \frac{\partial g}{\partial t}$ is the Jacobin of Eq. (6.6), $\Upsilon_{t_k|t_{k-1}}$ and $S_{t_{k-1}}$ are the covariance matrices of the predicted time distribution and noise term, respectively.

Note that the output of diagnosis at the time instant of the current event t_d is a battery SOH state distribution defined on the state axis, which cannot be used in LS-based prognosis. An initial distribution of operation time is needed for prognosis. To obtain the time distribution on the current Lebesgue state, the state distribution is converted to a time distribution, as shown in Fig. 6.3. In this conversion, the time instant of the current event t_d is set to be the mean of the time distribution μ_t, the sigma point $\mu_x - \sigma_x$ of the state distribution has a cross point d_c with the filtered battery SOH curve, marked in Fig. 6.3. The coordinates of the cross point ($\mu_t - \sigma_t$, $\mu_x - \sigma_x$) represent that sigma point $\mu_x - \sigma_x$ reaches the threshold at time instant $\mu_t - \sigma_t$. The time interval between t_d and d_c in Fig. 6.3 is set to be the variance of the time distribution. By this method, the variance of state distribution σ_x is approximately converted to that of the time distribution σ_t. Here, μ_x, σ_x, μ_t, and σ_t are the mean and variance of the state estimation, the mean and variance of the time distribution, respectively.

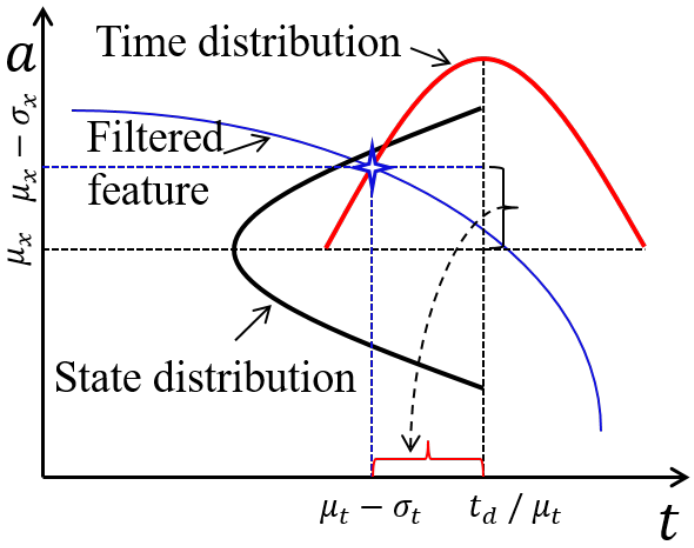


Figure 6.3: Conversion from state distribution to time distribution

6.1.3 LS-EKF for SOC diagnosis

Different methods were used to estimate the SOC, such as open-circuit voltage (OCV) method [112–115], and Coulomb-counting method [80, 108]. OCV method, based on the measurement of the equilibrium voltage of the battery, is used to estimate SOC though a look-up table since there exists a one-to-one mapping between OCV and SOC. However, the measurement of OCV requires a long relax time for battery achieving an equilibrium, which makes it infeasible for online application. Coulomb-counting method is usually chosen for its simplicity and less requirements on computation, which compares the amount of delivered electrical charge and the total capacity. The value of SOC P_{SOC} can be obtained as:

$$P_{SOC} = \frac{Q_t}{Q_0} \times 100\% \tag{6.8}$$

where Q_t is the remaining capacity at current time instant and Q_0 is the total capacity. Assume the initial SOC value of the battery is P_{SOC_0}, Eq. (6.8) can be described as:

$$P_{SOC} = P_{SOC_0} - \frac{\int idt}{Q_0} \tag{6.9}$$

where i is the current, which is considered to be positive and negative for discharge and charge, respectively.

However, this method has two limitations. One is that it starts from an initial SOC, which must be known and accurate. The other one is that the error originating from the sensor accuracy, current, and sampling frequency will accumulate over time and this method does not have the ability to recover from an inaccurate SOC estimation. To increase the accuracy and robustness of SOC esitmation, a model-based SOC estimation method is developed [74], which utilizes the OCV as an internal variable and estimates the SOC directly from a battery equivalent circuit model (ECM).

This research employs a second order ECM model as shown in Fig. 6.4 [44, 116]. Compared to the first-order ECM [117], this second order model better describes the

battery dynamic characterized by hybrid pulse power characterization (HPPC) data. In this model, R_0 is the Ohmic resistance, $i(t)$ is the discharge (positive) or charge (negative) current, $V(t)$ is the measured voltage, V_{OCV} is the OCV of the battery, and V_{Dif} and V_{CT} are the voltage drop caused by the ion diffusion and the charge transfer, respectively. Note that R_{Dif} and C_{Dif} compose an RC circuit that describes the battery behavior caused by the ion diffusion. The second RC circuit formed by R_{CT} and C_{CT} represents the charge transfer process in the battery.

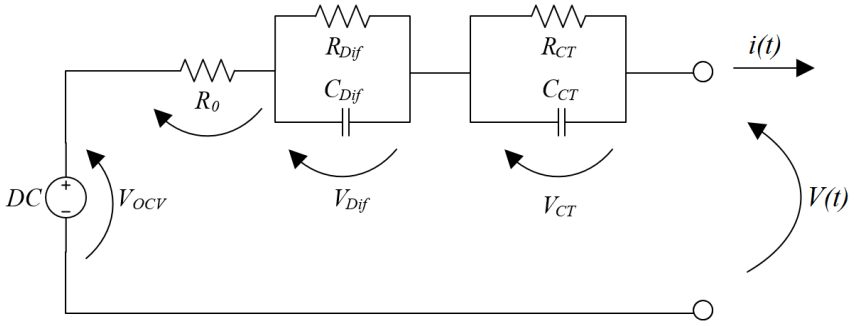


Figure 6.4: Second order ECM for Lithium ion batteries [44].

Based on this ECM model and the SOC achieved by Coulomb counting method, the state space model for SOC estimation in discrete time domain is given as:

$$\begin{cases} P_{SOC}(t_{k+1}) = P_{SOC}(t_k) - \frac{1}{Q_0} \cdot \int_{t_k}^{t_{k+1}} i(t)dt \\ V_{CT}(t_k) = e^{-\frac{\Delta t}{\tau_{CT}}} \cdot V_{CT}(t_k - \Delta t) \\ \qquad\qquad + R_{CT} \cdot i(t_k - \Delta t) \cdot \left(1 - e^{-\frac{\Delta t}{\tau_{CT}}}\right) \\ V_{Dif}(t_k) = e^{-\frac{\Delta t}{\tau_{Dif}}} \cdot V_{Dif}(t_k - \Delta t) \\ \qquad\qquad + R_{Dif} \cdot i(t_k - \Delta t) \cdot \left(1 - e^{-\frac{\Delta t}{\tau_{Dif}}}\right) \end{cases} \tag{6.10}$$

where Δt is the sampling period, and $\tau_{CT} = R_{CT}C_{CT}$ and $\tau_{Dif} = R_{Dif}C_{Dif}$ are the time constants of the two RC circuits in the ECM model. Note that this model is only calculated when the measured voltage reaches a new Lebesgue state in LS framework.

The relationship between the output voltage V and the open-circuit voltage V_{OCV}

is given as:

$$V_{OCV}(t_k) = V(t_k) + V_{CT}(t_k) + V_{Dif}(t_k) + R_0 \cdot i(t_k) \tag{6.11}$$

where R_0 is the battery internal ohmic resistance. The parameters in Eq. (6.10) and (6.11) can be identified by HPPC test [118] at the beginning of service life of the battery.

The state vector is defined as $x(t_k) = [P_{SOC}(t_k)\ V_{CT}(t_k)\ V_{Dif}(t_k)]^T$. The state space model of the battery can be written as:

$$x(t_{k+1}) = A \cdot x(t_k) + B \cdot u(t_k) \tag{6.12}$$

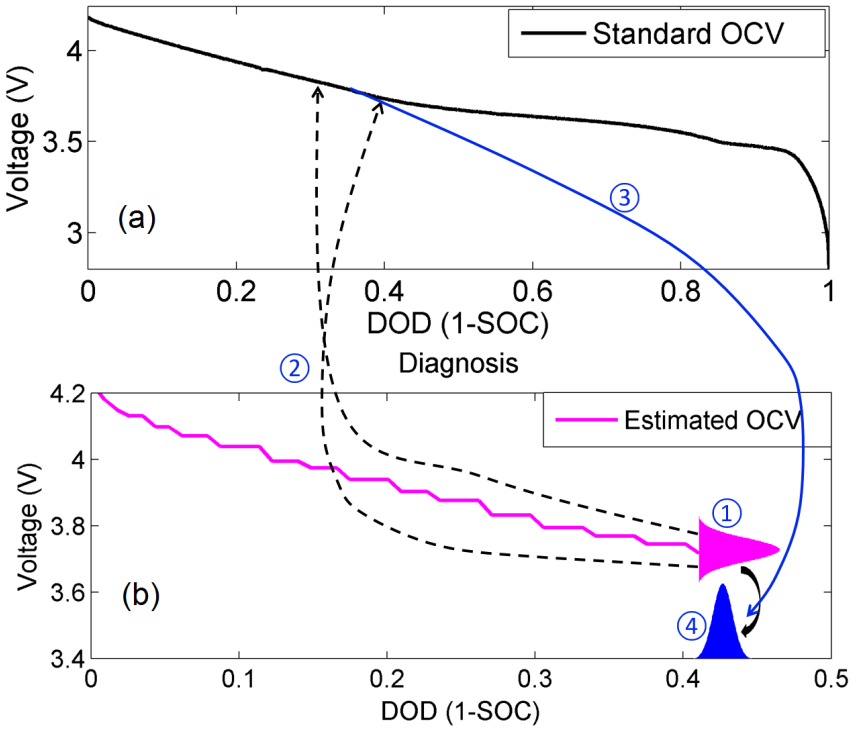
where input $u(t_k)$ is the charge or discharge current $i(t_k)$, and A and B matrices are defined as:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-\frac{\Delta t}{\tau_{CT}}} & 0 \\ 0 & 0 & e^{-\frac{\Delta t}{\tau_{Dif}}} \end{pmatrix}; B = \begin{pmatrix} -\frac{\Delta t}{Q_0} \\ R_{CT}\left(1 - e^{-\frac{\Delta t}{\tau_{CT}}}\right) \\ R_{Dif}\left(1 - e^{-\frac{\Delta t}{\tau_{Dif}}}\right) \end{pmatrix} \tag{6.13}$$

The non-linearity of the model is in the OCV output calculated by Eq. (6.11), in which OCV is not linear with respect to state $x(t_k)$.

The LS-EKF SOC diagnosis is implemented based on Eq. (6.11). With the estimation of OCV from Eq. (6.11), the SOC can be obtained from OCV-SOC look-up table. This involves the conversion from OCV distribution to SOC distribution as shown in Fig. 6.5. First, the V_{OCV} is estimated in LS-EKF by using measured voltage and model (6.11). The estimated OCV-DOD curve, shown in Fig. 6.5 (b) is compared with the standard OCV-DOD curve shown in Fig. 6.5 (a) (obtained at the beginning of the battery service) to get the estimation of SOC. Here $DOD = 1 - P_{SOC}$. As the estimated OCV is a Gaussian distribution in LS-EKF, the estimation of SOC is also subject to a Gaussian distribution.

First, the mean (μ_{OCV}) and 95% CI of OCV (CI_{OCV}) are achieved. From these values, the corresponding SOC values of CI_{OCV} can be obtained on the standard



①: calculate CI_{OCV} in LS; ②: calculate the corresponding CI_{SOC};
③: build a Gaussian pdf for SOC based on CI_{SOC};
④: get the SOC pdf.

Figure 6.5: SOC pdf conversion converted from estimated OCV pdf.

OCV-SOC curve shown in Fig. 6.5 (a). The values of CI_{OCV} are used to build a pdf of SOC, as shown in Fig. 6.5 (b).

6.1.4 LS-EKF for SOC prognosis

The SOC prognosis is conducted based on the first equation in (6.10), in which the discharge current $i(t)$ is the average of the future loading in the predefined working profile. The initial SOC value is achieved by Eq. (6.11) and the mapping in Fig. 6.19. By this method, an estimated RUL of SOC is obtained.

6.1.4.1 Parameter identification

The parameters of the second order ECM model need to be identified to estimate SOC based on Eq. (6.13).

The value of the RC parameters are identified based on the HPPC test, which is shown in Figure 6.6. The least squares method is used to identify the parameters, which minimize the sum of square error between the measured voltage and the voltage calculated by models (6.10) and (6.11). The dependence of parameters on SOC is negligible in this paper, so the data collected from SOC=50% is chosen to identify the parameters. The identified parameters are shown in Eq. (6.14), which will be used in OCV-SOC estimation based on Eq. (6.10) and Eq. (6.11).

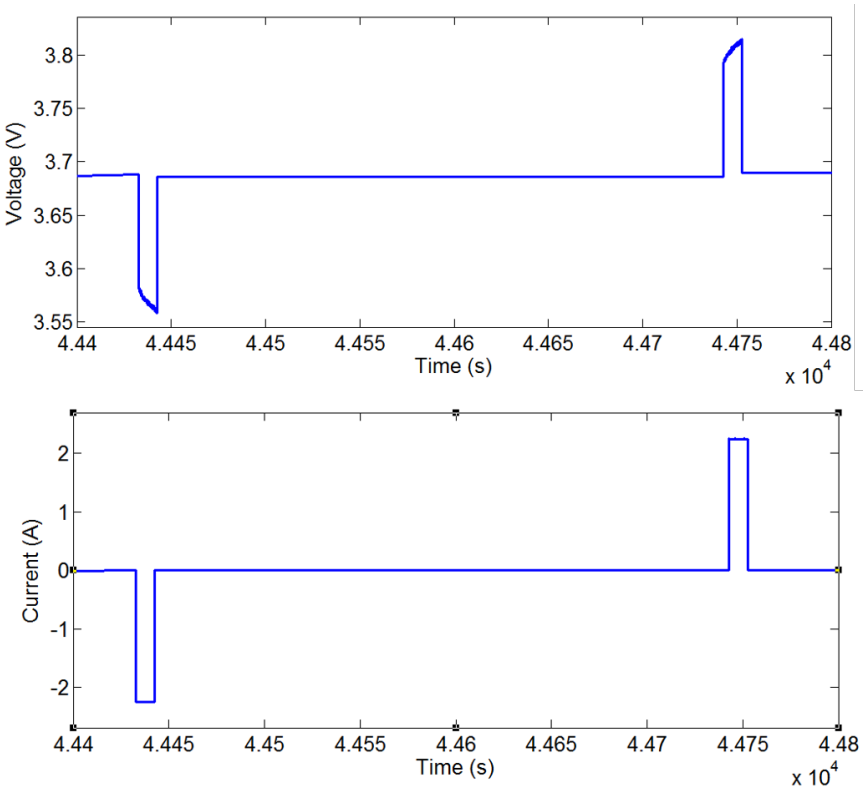


Figure 6.6: The collected HPPC data for a fresh battery at room temperature.

$$\begin{cases} R_{CT} = 11.7\ mW \\ \tau_{CT} = 7.1547\ s \\ R_{Dif} = 20.5\ mW \\ \tau_{Dif} = 7.0459\ s \\ R_0 = 0.0447\ \Omega \end{cases} \tag{6.14}$$

6.2 Experimental Results of SOC and SOH estimation

In this section, the proposed methods are demonstrated with an application to the SOC and SOH estimation of the Sony high drain 18650 Lithium-ion batteries with 2.25 Ah rated capacity. In this experiment, the capacity degradation is tested by Arbin BT2000 system under room temperature at a discharge current of 2.25 A. The battery capacity degradation speed is the decrease of the capacity with respect to charge-discharge cycles, which is collected by Coulomb counting method and shown in Figure 6.7. The failure threshold for the SOH is set to be 0.35 Ah and the battery capacity reaches this threshold at the 928th and 822th, respectively. From this series of experimental data, an empirical model will be established and will be used in the diagnosis and prognosis.

The SOC data is collected during each charge-discharge cycle of the battery, some examples of the collected data at different life stage of the battery are shown in Figure 6.8. The horizon axis is the discharged capacity during one cycle, the vertical axis is the measured voltage during the test. With the increase of the battery service time, the battery capacity degrades. The blue, red, and black curves show the capacity degradation trend in the initial state, middle stage, and late stage of battery service life.

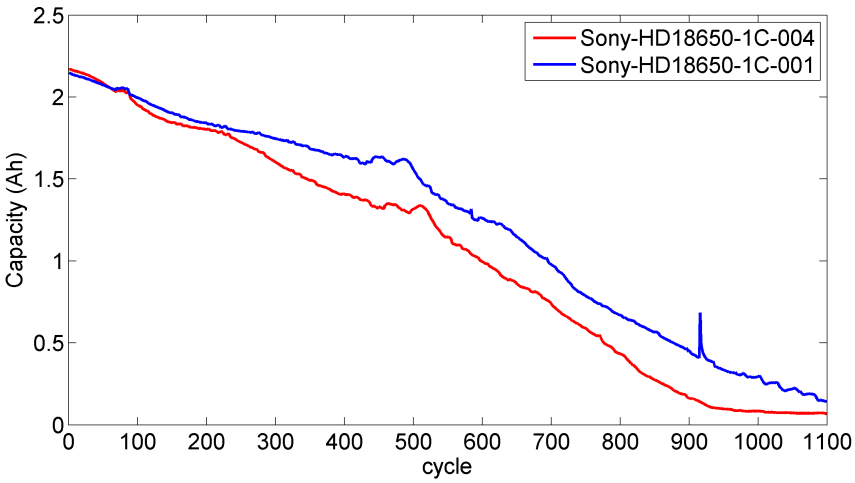


Figure 6.7: The capacity degradation curves of two batteries.

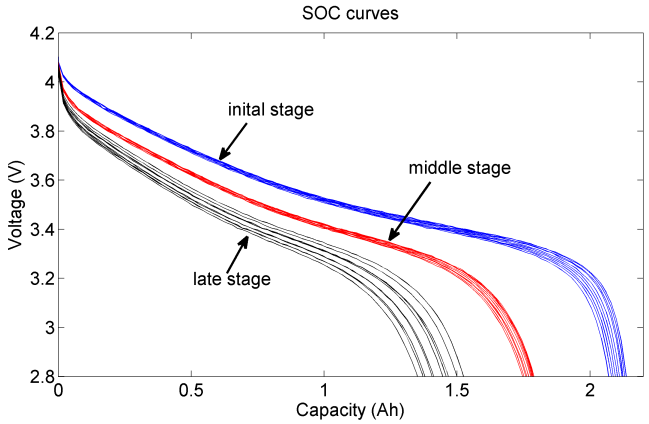


Figure 6.8: SOC curves in different battery service life stages.

6.2.1 SOH estimation and RUL prediction

6.2.1.1 Results achieved by RS-EKF

In traditional Riemann sampling framework, samples are taken equivalently along the time axis. The diagnosis and prognosis model is identically developed and is written in Eq. (3.12), which is re-written in Eq. (6.15) to refresh the memory.

$$C(t+1) = C(t) - \gamma \cdot \left| p_1 \cdot \left(p_2 + p_3 \cdot t + p_4 \cdot t^2 \right) \right|^{p_5} + \omega_C(t) \tag{6.15}$$

where $p = [1e^{-5}, 80, 0.08, -0.0008, 0.185]$, and the noise term $\omega_C(t)$ is required to be Gaussian.

To illustrate the proposed algorithms, Sony-HD18650-1C-004 battery is used as an example. Figure 6.9 shows the diagnostic results at the 400th cycle. The mean of capacity estimation is 1.4269 and the 95% confidence interval (CI) is [1.4080, 1.4459]. The upper sub-figure is the comparison of the capacity from Arbin system against the capacity estimation from EKF. The bottom sub-figure shows the comparison of initial baseline probability distribution function (pdf) (green) compared with the real-time estimated pdf (magenta) at the 400th cycle. Note that the diagnostic algorithm is executed 400 times in the past 400 cycles, *i.e.*, every time when a new measurement becomes available.

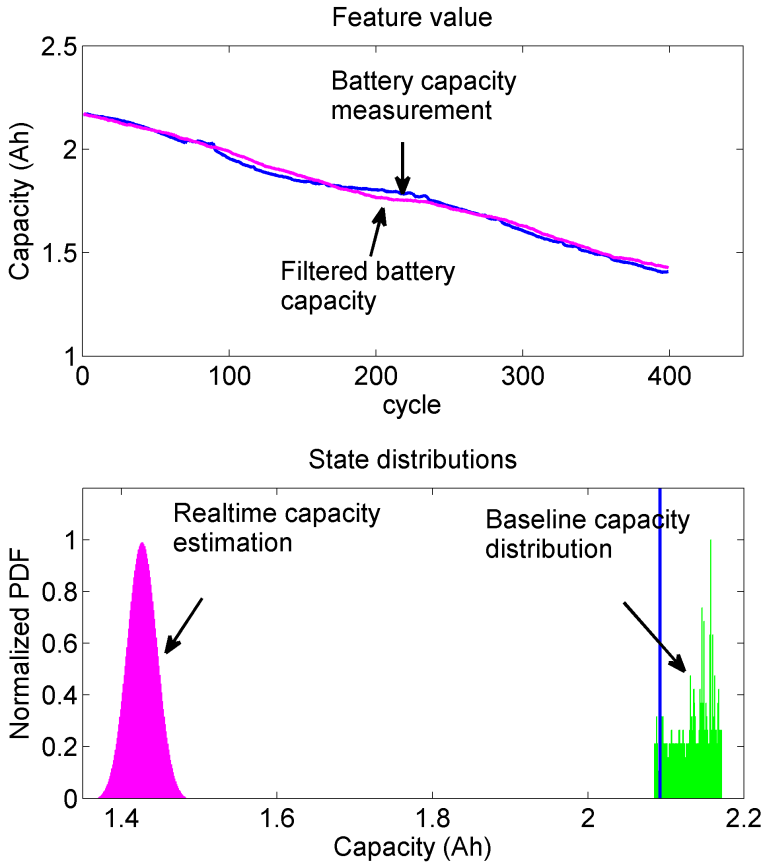


Figure 6.9: RS-EKF diagnosis for battery at the 400th cycle for Sony-HD18650-1C-004 battery.

With an estimation of the current battery capacity as the initial condition, the prognosis is executed to conduct the long-term prediction and estimation of RUL.

Figure 6.10 shows the expected value, upper and lower bounds of 95% CI of the battery capacity pdf at each future cycle.

The TTF distribution from EKF-based prognosis is a Gaussian distribution with mean value of 806 cycles when the mean predicted capacity reaches the failure threshold. The upper-bound and lower-bound of the distribution are calculated based on Eq. (6.4) in EKF algorithm, and can be simply approximated as the time instant when the upper-bound and lower-bound of the capacity distribution reach the failure threshold, details are shown in Figure 6.10. By this means, the standard deviation of the TTF distribution is approximated to be 18.5 cycles.

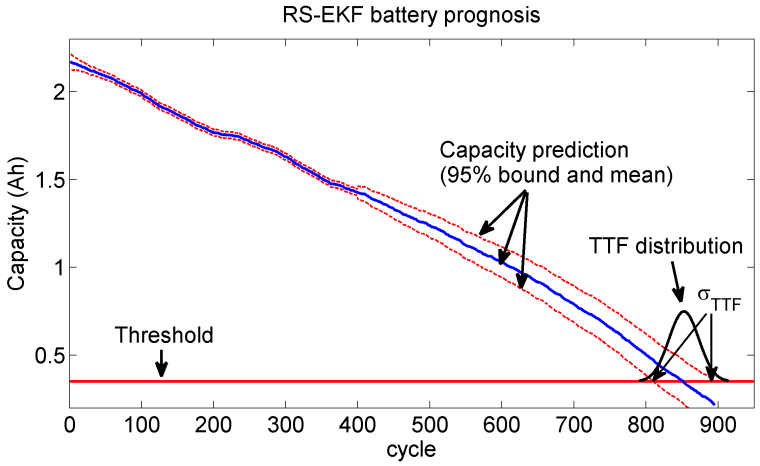


Figure 6.10: RS-EKF prognosis for battery at the 400th cycle.

In this figure, the predicted RUL is 406 cycles. The distance between the prediction and ground truth is 16 cycles. The 95% CI of the RUL pdf is [769 843], which indicates that the uncertainty accumulated along the prediction horizon is very large.

6.2.1.2 Results achieved by LS-EKF

To implement LS-FDP for the battery capacity degradation, initially, 40 uniformly distributed Lebesgue states are defined in the battery's full capacity of 2.25 Ah. With this setting, the diagnostic algorithm is executed only when the capacity degrades

from one Lebesgue state to another. During the diagnosis process, the length of the Lebesgue states is optimally adjusted according to the fault growth speed. If the fault grows faster, the next Lebesgue length will be decrease, otherwise, the next Lebesgue length will be increased. The adjustment details are illustrated in Chapter 5. The number of Lebesgue states changes during this optimization process.

In Lebesgue sampling framework, an diagnostic model is developed, which is given in Eq. (5.4). The model is re-written here (Eq. (6.16)) to refresh the memory. Note that the noise term in Eq. (6.16) is required to be Gaussian.

$$C(t_{k+1}) = C(t_k) - p_d \cdot C(t_k) \cdot D(t_k) \cdot sgn(C(t_k) - C(t_{k-1})) + \omega_C(t_k) \tag{6.16}$$

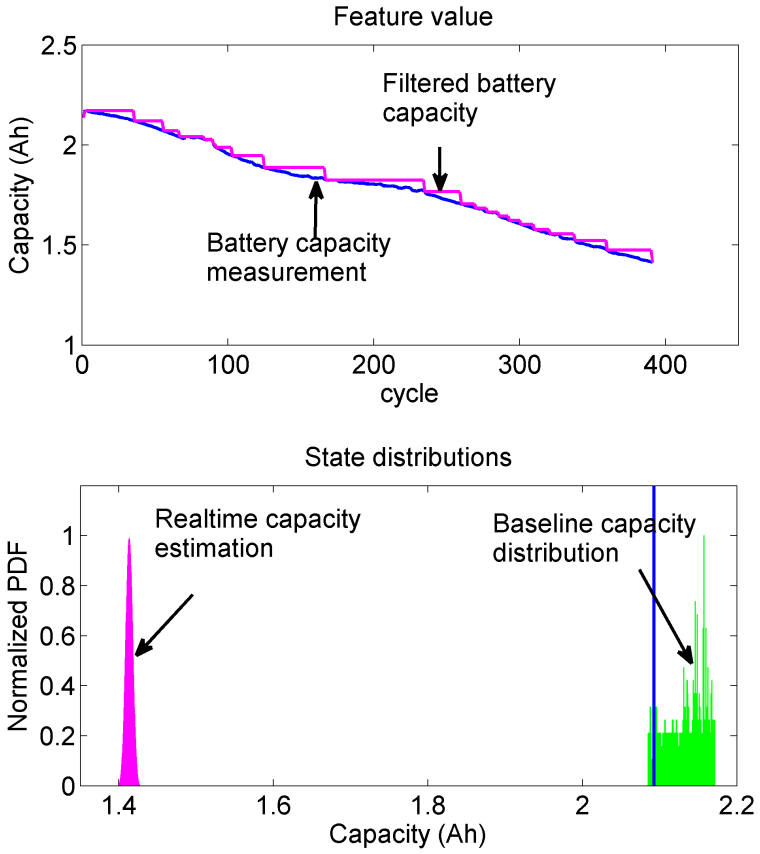


Figure 6.11: LS-EKF diagnosis for battery at the 400th cycle.

Figure 6.11 shows the diagnostic results at the 400th cycle based on LS-EKF. The mean of capacity estimation is 1.414 and the 95% CI is [1.409, 1.414]. Same as

the example in RS-EKF, the upper sub-figure shows the comparison of capacity from Coulomb counting (blue) against the estimated mean value from diagnosis (magenta). The lower sub-figure shows the comparison of initial baseline pdf compared with the real-time estimated pdf at the 400th cycle. Note that the diagnostic algorithm is only executed 25 times in the past 400 cycles.

The RUL prediction is conducted by propagating the operating time distribution at each Lebesgue state directly, among which the one on the Lebesgue state that equals to the failure threshold is the RUL probability distribution function (pdf). The prediction model is given in Eq. (5.5) and copied as Eq. (6.17). The noise term $\omega_k(t_k)$ is Gaussian, which is different from the requirement in Chapter 5.

$$t_{k+1} = t_k + p_p \cdot C(t_k) \cdot D(t_k) \cdot exp\left(-\dot{C}(t_k)\right) + \omega_k(t_k) \tag{6.17}$$

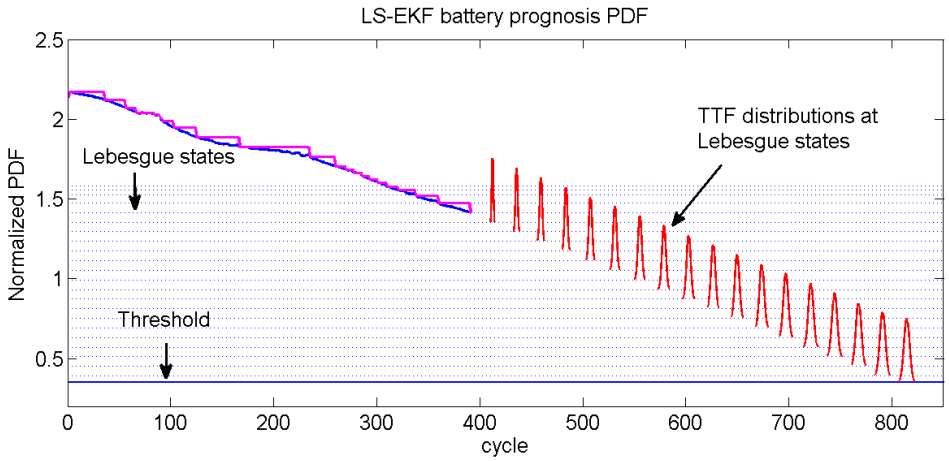


Figure 6.12: LS-EKF prognosis for battery at the 400th cycle.

Figure 6.12 shows the prognostic results at the 400th cycle. To make the figure clear, only the time distribution pdf at a few selected Lebesgue states are plotted. The prediction horizon is 20 Lebesgue states, which is very small compare to the 444 cycles in RS-EKF. The predicted TTF for this battery is 814.35 and the RUL is 414.35 cycles. The 95% CI of the TTF distribution is [809.1 819.6]. The uncertainty is much smaller than that of RS-based prognosis due to the small prediction horizon.

Compared with the ground truth TTF of 822,the difference between ground truth and the prediction is 7.65 cycles.

6.2.1.3 Comparison of RS-EKF and LS-EKF

Compared to RS-EKF prognosis with large horizon (444 cycles), the LS-EKF prognosis shown in Figure 6.12 only has a prognostic horizon of 20 Lebesgue states. The reduction of computation time is (0.083668-0.005553)/0.083668=93.36% and the computation is about 15.06 times faster.

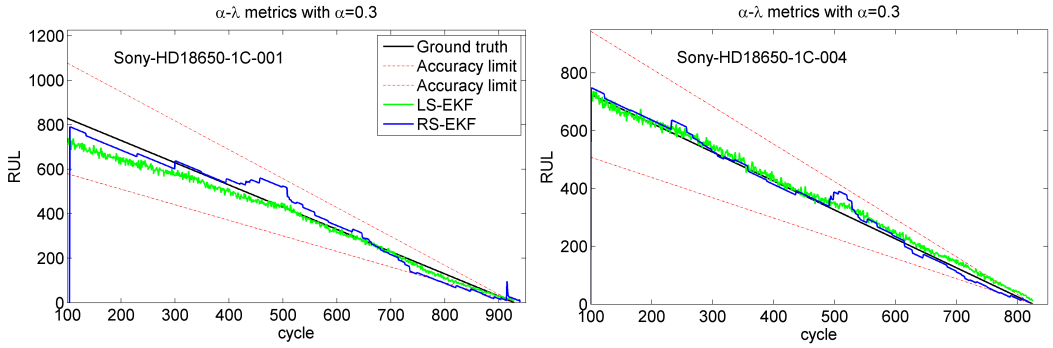
Diagnostic and prognostic results of RS-EKF and LS-EKF algorithms are compared in Table 6.1. Compared with RS-EKF prognosis with a horizon of 444 cycles at the 400th cycle, the LS-EKF prognosis has a horizon of 20 Lebesgue states. The computation time for every LS-EKF prognosis routine is 0.005553 s, which is only 6.64% of that of the RS-EKF prognosis (0.083668 s). Note that the Lebesgue state length in the LS-EKF prognosis is changed according to the fault growth speed to keep a closer monitoring on the SOH. If the fault growth speed becomes faster along the prediction steps, the Lebesgue state length for the following prediction steps will be decreased, otherwise, it will be increased. The computational sources is optimally distributed during the diagnosis and prognosis process by increasing the Lebesgue state length and reducing the unnecessary execution when the fault grows slowly. When the fault grows fast, more computational sources is assigned to the FDP algorithm to monitor the health state of the system.

Accuracy is one of the most important properties in FDP. In order to compare the accuracy of RS-EKF and LS-EKF methods, α - λ matrix is introduced as shown in Figure 6.13 with $\alpha = 0.3$. It is clear from Figure 6.13 that the mean of the predicted RUL for RS-EKF is as accurate as that of LS-EKF. However, the variance of predicted RUL of LS-EKF is much smaller as shown in Table 6.1, which is the natural benefit from Lebesgue sampling methodology, since the prediction horizon in

Table 6.1: Comparison of Traditional RS-EKF and LS-EKF for Battery

Diagnosis results	RS-EKF	LS-EKF
Capacity expectation	1.4269	1.414
Capacity 95% CI	[1.4080 1.4459]	[1.409 1.414]
Execution numbers	400 (100%)	25 (6.25%)
Prognosis results	RS-EKF	LS-EKF
True TTF	822	822
Estimate TTF	806	814.35
95% CI of TTF	[769 843]	[809.1 819.6]
Prognostic horizon	444	20
Computation time (s)	0.083668 (100%)	0.005553 (6.64%)

LS-EKF is much smaller than that of RS-EKF, the uncertainty accumulation during the prediction process is much smaller. Based on these advantages, LS-EKF can provide strong support for decision-making. More importantly, the LS-EKF required less calculation sources compared with RS-EKF, which makes it more feasible for distributed FDP with limited computational sources.



Note that the peak in the mean value of the predicted RUL in the left sub-figure, which is caused by the peak in the measurement, as shown in Figure 6.7.

Figure 6.13: Prediction accuracy comparison between RS-EKF and LS-EKF.

6.2.2 SOC estimation

A standard OCV-SOC is obtained by discharging the battery at a low current. The batteries are fully charged and discharged under room temperature with a constant current of 0.05 C. The measured voltage can be used to approximate V_{OCV} since the

discharging current is very small, the voltage drop on the internal impedance and two RC circuits is negligible. The measured voltage and SOC curve is shown in Figure 6.14, which is also the approximated $V_{OCV} - DOD(Depth\ of\ Discharge)$ curve. Note that $DOD = 1 - P_{SOC}$. The SOC is achieved by Coulomb counting method.

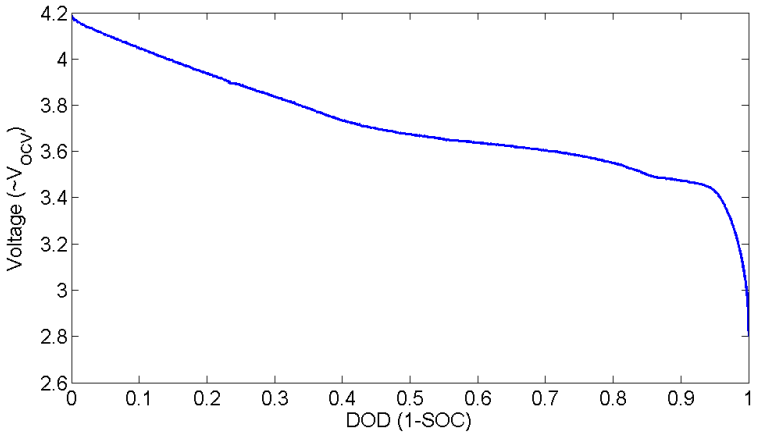


Figure 6.14: The relation of OCV and SOC collected with discharge current of 1/20C.

As shown in Figure 6.8, the discharge capacity decreases with the increasing of the battery service time. Thus, in the SOC estimation, the initial battery capacity during each cycle Q_0 is updated based on the posterior pdf from the SOH estimation results in Section 6.2.1.2, which can accommodate the influence of battery capacity degradation during the service life. The SOC estimation is conducted with RS-EKF and LS-EKF algorithms, which are discussed in the following parts.

6.2.2.1 SOC estimation by RS-EKF

OCV-SOC estimation results are achieved by RS-EKF based on models (6.10) and (6.12), Figure 6.15 shows the results at (2.1454-0.8618) Ah/2.1454 Ah = 60% SOC. The mean of OCV is 3.7313 V and the 95% CI is [3.7031, 3.7595] V. The corresponding CI of SOC can be found in Figure 6.14 by checking with the standard OCV-SOC mapping, as shown in Figure 6.16. First, the CI of OCV CI_{OCV} is achieved, and then the corresponding SOC points of CI_{OCV} is found on the standard OCV-SOC

curve. The values are used to build a pdf of SOC, as shown in blue in Figure 6.16. CI_{DOD} and CI_{SOC} are [0.3752, 0.4407] and [0.5593, 0.6248], respectively. The upper sub-figure is the comparison of the voltage measured by Arbin system against the voltage estimated by RS-EKF. The magenta curve is OCV-SOC curve obtained by Eq. (6.11), in which V_{CT} and V_{Dif} are calculated by Eq. (6.10). The blue curve is the measured voltage. The bottom sub-figure shows the comparison of initial baseline pdf of OCV (green) compared with the real-time estimated OCV pdf (magenta) at 60% SOC. Note that the diagnostic algorithm is executed 50 times in the past 50 sampling points, *i.e.*, every time when a new measurement becomes available.

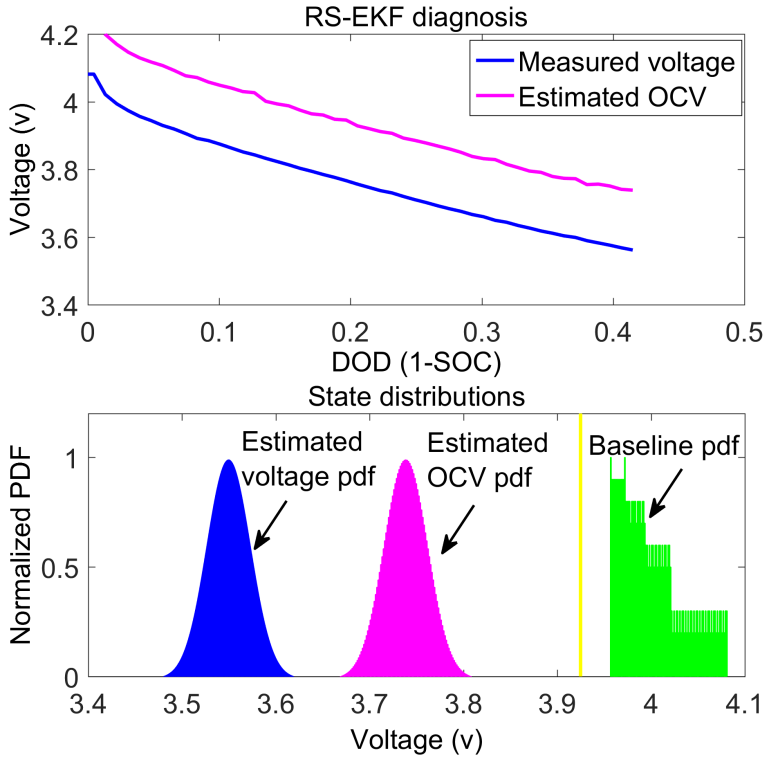


Figure 6.15: Estimated OCV-SOC results by RS-EKF.

With an estimation of the current battery SOC as the initial condition, the prognosis is executed to conduct the long-term prediction and estimation of RUL with the assumption that the future discharge current is constant. Figure 6.17 shows the expected value, upper and lower bounds of 95% CI of the battery capacity pdf

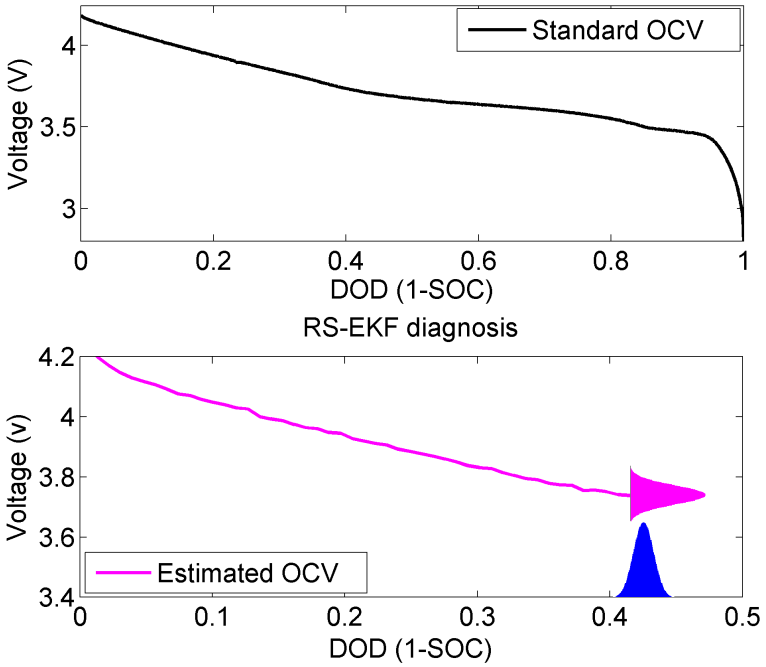
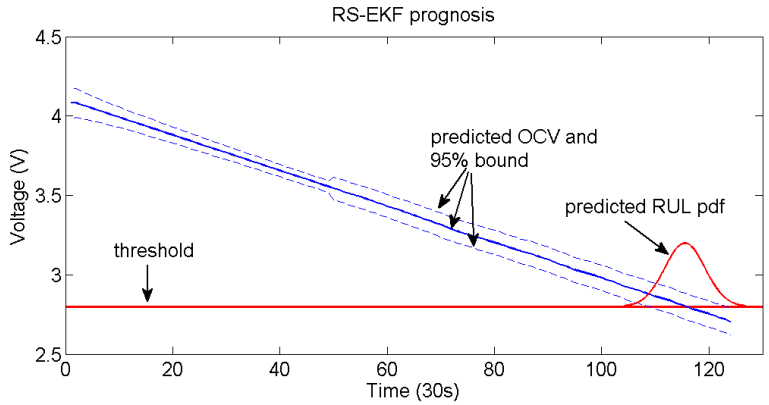


Figure 6.16: SOC pdf conversion converted from estimated OCV by RS-EKF.

at each future cycle. The predicted TTF for this battery is $115.5 \times 30\ s = 3465\ s$ and the RUL is $65.5 \times 30\ s = 1965\ s$. The unit for the horizontal axis is the index of sampling data points with a sampling period of 30 s in this study. The 95% CI of the TTF distribution is [108 123]. The prediction horizon is 74. Compared with the ground truth TTF of 116, the difference between ground truth and the prediction is $0.5 \times 30\ s = 15\ s$.



Note that the unit of horizontal axis in this figure is the sampling period (30 s), which means the data are sampled every 30 s.

Figure 6.17: SOC prognosis based on RS-EKF at 60% SOC.

6.2.2.2 SOC estimation by LS-EKF

SOC estimation model in LS-EKF is developed based on Eq. (6.10) and Eq. (6.11), as given in Eq. (6.18).

$$\begin{aligned} V_{OCV}(t_{k+1}) =& V(t_k) - D(t_k) \cdot sgn(V(t_k) - V(t-k-1)) \\ &+ V_{Dif}(t_k) + V_{CT}(t_k) + R_0 \cdot i(t_k) + \omega_V(t_k) \end{aligned} \tag{6.18}$$

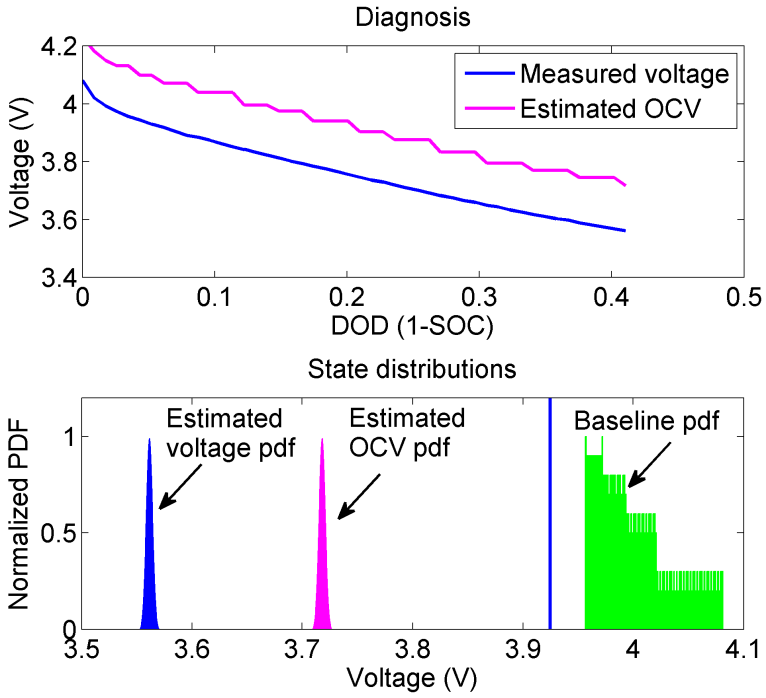


Figure 6.18: Estimated OCV-SOC results by LS-EKF.

Figure 6.18 shows the diagnostic results at 60% SOC based on LS-EKF. The mean of OCV estimation is 3.72 V and the 95% CI is [3.7156, 3.7244] V. CI_{DOD} and CI_{SOC} are achieved with the same means as Figure 6.16. As shown in Figure 6.19, the results are [0.4115, 0.4251] and [0.5749, 0.5885], respectively. The upper sub-figure shows the comparison of voltage from measurement (blue) against the estimated mean value from diagnosis (magenta). The lower sub-figure shows the comparison of initial baseline pdf compared with the real-time estimated pdf at the 60% SOC. Note that the diagnostic algorithm is only executed 18 times in the past 50 sampling data points, which saves 65.8% computation resources.

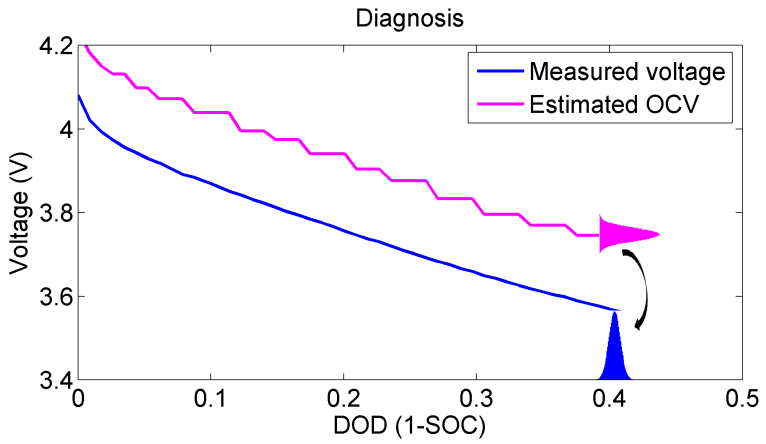


Figure 6.19: SOC pdf conversion converted from estimated OCV by LS-EKF.

The OCV-SOC curve tested under room temperature with a discharging current of 1/20 C and the diagnosis results for the whole OCV-SOC curve based on RS-EKF and LS-EKF are shown in Figure 6.20. The estimated OCV-SOC curve by RS-EKF (green) and LS-EKF (magenta) is close to the standard OCV-SOC curve (black) compared with the measured V-SOC curve (blue). The root mean square (RMS) error between the estimated and standard OCV-SOC curve is 0.05610 and 0.0601, respectively. The RMS error between the V-SOC curve and standard OCV-SOC curve is 0.224, which indicates an accurate OCV-SOC curve is achieved by the proposed RS-EKF and LS-EKF method.

The SOC prognostic in LS-EKF is conducted based on the following model:

$$t_{k+1} = t_k + p_p(t) \cdot P_{SOC}(t_k) \cdot D(t_k) \cdot exp\left(-\dot{P}_{SOC}(t_k)\right) + \omega_t(t_k) \tag{6.19}$$

where $p_p(t)$ is influenced by the discharge current. The OCV-SOC data used in this Chapter is collected under a constant discharge current, so p_p is a constant, which is set to be 200. The model noise ω_t is Gaussian.

Figure 6.21 shows the prognostic results at the 60% SOC. To make the figure clear, only the time distribution pdf at a few selected Lebesgue state are plotted. The prediction horizon is 23 Lebesgue states. The predicted TTF for this battery is $107.18 \times 30\ s = 3215.4\ s$ and the RUL is $56.18 \times 30\ s = 1685.4\ s$. Note that

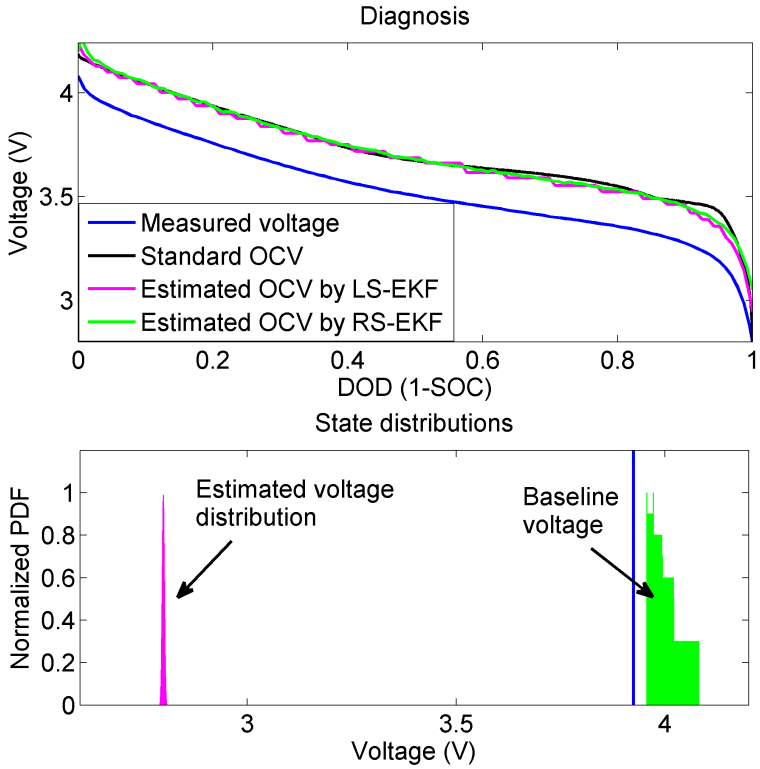
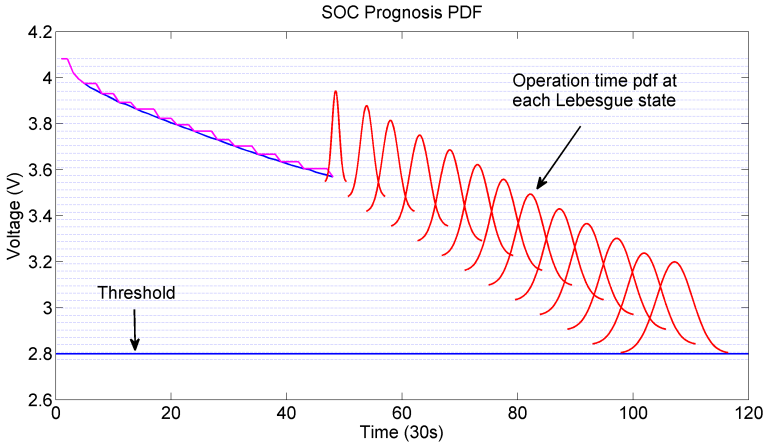


Figure 6.20: The comparison between standard and estimated OCV-SOC curve by LS-EKF.



Note that the unit of horizontal axis in this figure is the sampling period (30 s), which means the data are sampled every 30 s.

Figure 6.21: LS-EKF based SOC prognosis for battery at 60% SOC.

the unit on the horizontal axis is the index of sampling data points with a sampling period of 30 s in this study. The 95% CI of the TTF distribution is [100.99 113.37]. The uncertainty is much smaller than that of RS-based prognosis due to the small

prediction horizon. Compared with the ground truth TTF of $116 \times 30\ s = 3480\ s$, the difference between ground truth and the prediction is $8.82 \times 30\ s = 264.6\ s$.

Same as the results in Section (6.2.1.3), LS-EKF prognosis has a smaller prediction horizon (23 Lebesgue states) compared with RS-EKF (74 data points with a sampling period of 30 s). The reduction of computation time is (0.012484-0.004582)/0.012484=63.3% and the computation is about 2.72 times faster.

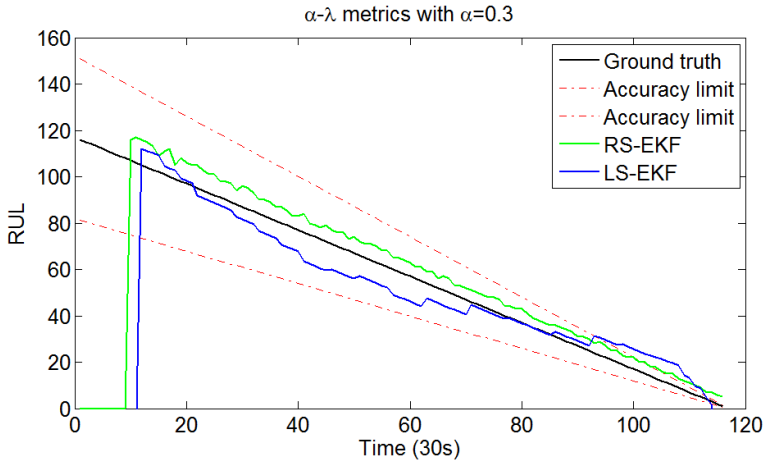
6.2.2.3 Comparison of SOC estimation by RS-EKF and LS-EKF

Diagnostic and prognostic results of SOC based on RS-EKF and LS-EKF algorithms are compared in Table 6.2. Compared with RS-EKF prognosis with a horizon of 74 at 60% SOC, the LS-EKF prognosis has a horizon of 23 Lebesgue states. The computation time for every SOC prognosis by LS-EKF is 36.7% of that of the RS-EKF prognosis. Note that the Lebesgue state length in the LS-EKF prognosis is changed according to the fault growth speed to optimize the distribution of computation resources.

Table 6.2: Comparison of RS-EKF and LS-EKF for SOC Estimation

Diagnosis results	RS-EKF	LS-EKF
OCV expectation	3.7313	3.72
OCV 95% CI	[3.7031 3.7595]	[3.7156 3.7244]
Execution numbers	50 (100%)	18 (36%)
Prognosis results	RS-EKF	LS-EKF
True TTF	116	116
Estimate TTF	115.5	107.18
95% CI of TTF	[108 123]	[100.99 113.37]
Prognostic horizon	74	23
Computation time	0.012484 (100%)	0.004582 (36.7%)

In order to compare the accuracy SOC esitmation by RS-EKF and LS-EKF methods, α - λ matrix is shown in Figure 6.22 with $\alpha = 0.3$. Judging from Figure 6.22, the mean of the predicted RUL for RS-EKF is as accurate as that of LS-EKF. However, the variance of predicted RUL of LS-EKF is much smaller as shown in Table



Note that the unit of horizontal axis in this figure is the sampling period (30 s), which means each data point represents 30 s.

Figure 6.22: SOC prediction accuracy comparison between RS-EKF and LS-EKF.

6.2. Note that the unit on the horizontal axis is the the index of sampling data points with a sampling period of 30 s in this study. The LS-EKF required less calculation sources compared with RS-EKF, which makes it more feasible for distributed FDP with limited computational sources.

6.2.3 Parameter adaptation for SOC estimation

The parameter values in model (6.10) change with the degradation of the battery, which makes the value in Eq. (6.14) not constant during the whole life of the battery life. To illustrate its effects, the estimated OCV-SOC curve (red) is compared with the standard one (blue) in Figure 6.23, 6.24, and 6.25 at the 2nd, 200th, and 500th cycle. The black curves are estimated with the initial parameter settings (shown in Eq. (6.14)), it is clear that the estimated OCV-SOC curves diverge from the standard one with the degradation of the battery. In order to get an accurate SOC estimation with the degradation of the battery, the parameter values need to be updated accordingly. In our research, the parameters are updated based on the RLS method, which is discussed in Section 4.2.1. The estimated OCV-SOC curves achieved by the SOC estimation model with updated parameters are shown in red in

Figure 6.23, 6.24, and 6.25. With parameter adaptation, more accurate OCV-SOC estimation is achieved and the RMS error is reduced, as shown in Table 6.3. It is clear that the OCV-SOC estimation is improved by the proposed method, and this method is verified with the capability to handle OCV-SOC estimation at different survive life stages.

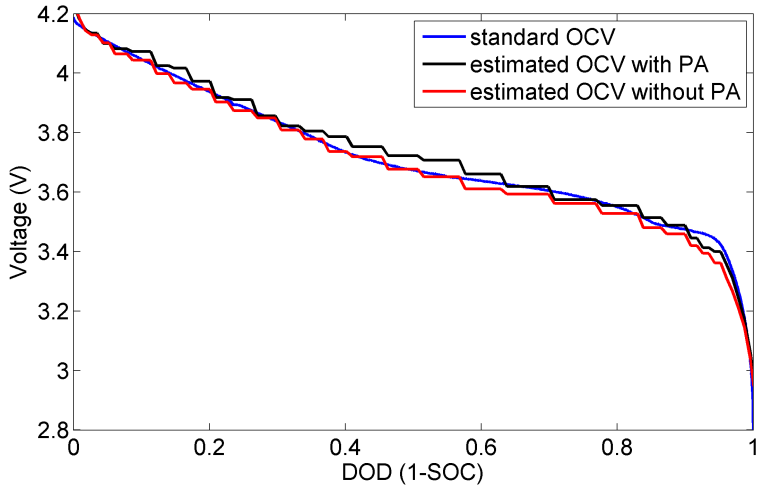


Figure 6.23: Estimated OCV-SOC curves at the 2nd cycle.

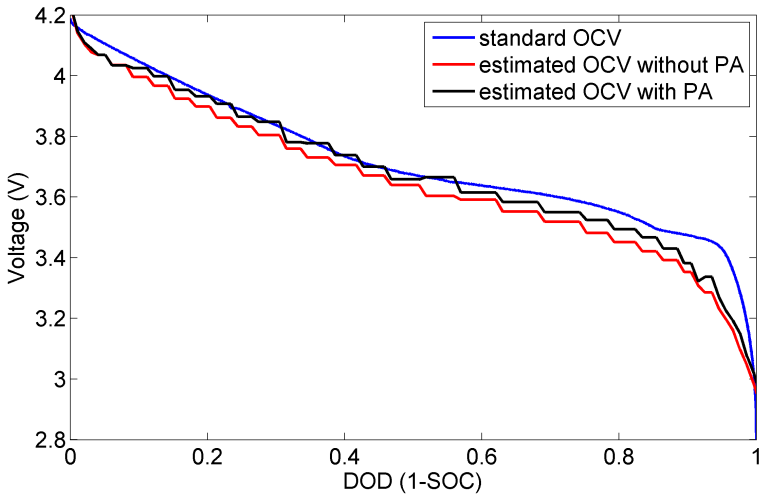


Figure 6.24: Estimated OCV-SOC curves at the 200th cycle.

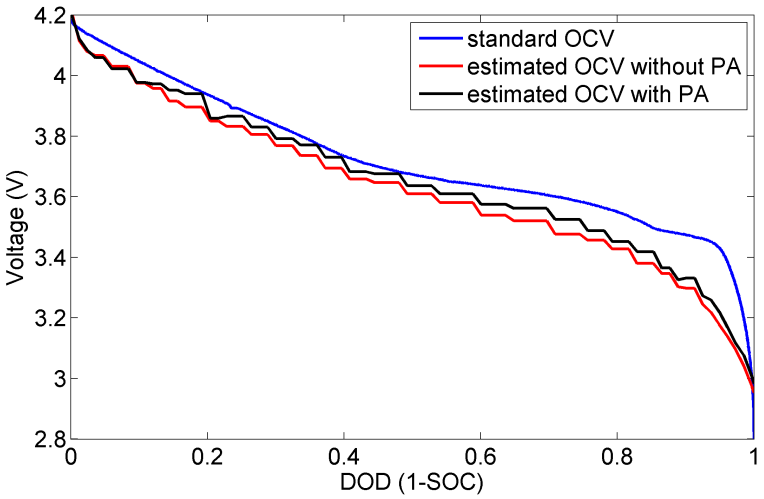


Figure 6.25: Estimated OCV-SOC curves at the 500th cycle.

Table 6.3: Root mean square error of estimated OCV with/without parameter adaptation.

	2nd cycle	200th cycle	500th cycle
RMS error with PA	0.0374	0.0841	0.1083
RMS error without PA	0.0608	0.1127	0.1343

6.3 Conclusions

In order to implement FDP algorithms in embedded system with limited computation resources to estimate the SOH and SOC of Lithium ion batteries, a new fault diagnosis and prognosis (FDP) methodology based on EKF is developed in Lebesgue sampling framework (LS-EKF). An experiment of Lithium-ion battery SOH and SOC estimation with comparison against traditional RS-based approach is presented. The estimated SOH is used to update the initial value in SOC estimation to achieve accurate results. The estimation of OCV-SOC curve is conducted based on a second order ECM model, the ECM model parameters are updated to accommodate the degradation of the battery, which is proved to be able to reduce the RMS error between the estimated OCV-SOC curve and the standard one. It is demonstrated that the proposed approach is able to reduce the requirement on computational sources

compared with traditional RS-EKF and provide reliable SOH and SOC estimation. This proposed approach combines the advantages of EKF and LS method, which results in low computation and small uncertainty accumulation.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

Fault diagnosis and prognosis (FDP) is employed to estimate the fault state, and maintain the reliability, productivity, safety, and availability of industrial systems. There are two configurations of FDP implementation: centralized and distributed. With the development of modern technologies, the complexity of system increases rapidly, distributed FDP design is widely accepted in engineering design, especially for complicated systems. With this trend, more and more FDP functions are deployed on local processors and embedded systems to alleviate the requirements on communication bandwidth, power, and computation. These local processors and embedded systems have very limited computational resources, which limits the application of traditional FDP algorithms. Traditional FDP algorithms are developed in Riemann sampling framework, which takes samples and executes algorithms in periodic time intervals and requires significant computational resources.

To overcome this bottleneck, we propose a Lebesgue sampling-based FDP (LS-FDP) framework with a philosophy of "execution when needed". In LS-FDP, the state axis is divided by a number of predefined states (also called Lebesgue states). The computation in LS-FDP will be triggered only when the measurement reaches a new Lebesgue state (an "event" happens) and the prognosis will be executed based on the LS-based model. With the feature of "execution only when necessary" in Lebesgue sampling (LS), the computation efforts in LS-FDP can be significantly reduced by

eliminating unnecessary computation when fault growth is slow. The algorithms are designed based on particle filter method (PF), and verified with an application to the state-of-health (SOH) of Lithium-ion batteries. Experimental comparison against RS-FDP shows that the proposed LS-FDP is able to reduce the overall computation time significantly without sacrificing the performance.

In order to optimize the computation resources, the Lebesgue length is adaptively adjusted according to the fault growth speed since when the fault growth becomes fast, more attention should be spent on the system monitoring. With this method, the FDP is executed in a low frequency when the fault growth is slow, and in a high frequency when the fault growth is fast. This enables the LS-FDP algorithms to monitor the fault state effectively and the computational resources are optimally distributed.

In order to improve the performance of the LS-FDP based on PF, parameter adaptation and uncertainty management based on recursive least square method are introduced. Parameter adaptation adaptively adjust the model parameters to accommodate the effects of varying operational and environmental conditions on fault growth, which produce a more accurate fault state estimation and remaining useful life (RUL) prediction. Uncertainty management analyzes and manages the uncertainties originate from model inaccuracy, measurement noise, process noise, and unknown future loading, which results in a precise time to failure (TTF) distribution and a trust-worthy decision for maintenance. The methods are verified with application to the SOH of Lithium-ion batteries. Experimental results for LS-FDP with/without parameter adaptation show that the accuracy of the RUL prediction is enhanced. The uncertainty of the predicted TTF is remarkably reduced by the proposed uncertainty management method, which is a strong support for the decision-making in the FDP system.

Extended Kalman filter (EKF) has much less requirements on computation and

has demonstrated performance in many applications. A more efficient algorithm is designed, which integrates Lebesgue sampling method and EKF (called LS-EKF) for state estimation and RUL prediction in nonlinear systems. An experiment of Lithium-ion battery SOH diagnosis and prognosis with comparison against traditional RS-based approach is presented. It is demonstrated that LS-EKF is able to reduce the requirement on computational sources compared with traditional RS-EKF. This proposed approach requires low computation and produces small uncertainty accumulation.

LS-EKF based algorithms are developed to estimate the SOC and SOH, in which the estimated SOH is used to adjust the initial condition during the SOC estimation. The OCV-SOC curves are estimated by a second order ECM model, the parameters for ECM are initialized by HPPC test and updated with the degradation of the battery. The algorithms are verified by the SOH degradation curve and OCV-SOC data. The experimental results show that the proposed LS-EKF is time efficient with less uncertainty accumulation.

7.2 Future Works

The verification and validation of our proposed methods are conducted based on the SOC collected under constant discharge current. In this case, the SOC degradation is a linear process. We are currently testing batteries and collecting OCV-SOC data with varying discharge current, the data will be used to verify and validate our algorithms in the future.

The parameters in OCV-SOC curve estimation changes with the degradation of Lithium ion batteries. In our research, a RLS method is used to update the parameters. The updated parameters need to be verified by the parameters achieved by the HPPC test data in the future.

In the LS-FDP prognosis process, we use a set of particles to approximate the

operation time distribution at each Lebesgue state. In order to arrive at a RUL distribution, all the particles need to be propagated until the failure threshold, which usually indicates a lot of computations. A mean value and the CI is proposed to represent the operation time distribution, which will be propagated based on the prognosis model to achieve the mean and CI of the RUL. This will further reduces the computation requirements during the prognosis process. However, the accuracy and precision of the achieved mean and CI need to be compared against the one from PF and EKF algorithms in both RS and LS framework.

PUBLICATIONS

1. **W. Yan**, B. Zhang, G. Zhao, J. Weddington, and G. Niu, "Uncertainty Management in Lebesgue Sampling-based Diagnosis and Prognosis for Lithium-ion Battery" , IEEE Transactions on Industrial Electronics, Accepted.

2. **W. Yan**, B. Zhang, W. Dou, D. Liu, and Y, Peng, "Low-Cost Adaptive Diagnosis and Prognosis with Li-ion Battery Application", IEEE Transactions on Automation Science and Engineering, PP(1)2017.

3. **W. Yan**, B. Zhang, X. Wang, W. Dou, and J. Wang, "Lebesgue Sampling-Based Diagnosis and Prognosis for Lithium-Ion Batteries" , IEEE Transactions on Industrial Electronics, 63(3), 2016.

4. **W. Yan**, B. Zhang, and M. Orchard, "Parameters Optimization of Lebesgue Sampling-based Fault Diagnosis and Prognosis with Application to Li-ion Batteries", In 2016 Annual Conference of the Prognostics and Health Management Society

5. **W. Yan**, M. Abdelrahman, B. Zhang, and Paul Ziehl. "Particle Filtering Based Structural Assessment with Acoustic Emission Sensing", In the Proceedings of 43rd Annual Review of Progress in Quantitative Nondestructive Conference (QNDE'16), 2016.

6. **W. Yan** and B. Zhang, "Extended Kalman filter development in Lebesgue sampling framework with an application to Li-ion battery diagnosis and prognosis", In the Proceedings of Third European Conference of the Prognostics and Health Management Society, July, 2016, Bilbao, Spain.

7. **W. Yan** and B. Zhang, "Uncertainty management in Lebesgue sampling-based fault diagnosis and prognosis", Proceedings of the ASME 2016 Pressure Vessels & Piping Conference (PVP'16), July, 2016, Vancouver, Canada.

8. **W. Yan**, W. Dou, D. Liu, Y. Peng, and B. Zhang, "Parameters Adaption of Lebesgue Sampling-based Diagnosis and Prognosis for Li-ion Batteries", In 2015 Annual Conference of the Prognostics and Health Management Society.

9. J. Weddington, **W. Yan**, W. Dou, and B. Zhang, "Battery Capacity Anomaly Detection and Data Fusion", In 2015 Annual Conference of the Prognostics and Health Management Society.

Bibliography

[1] S. Namburu, M. Azam, J. Luo, K. Choi, and K. Pattipati. "Data-Driven Modeling, Fault Diagnosis and Optimal Sensor Selection for HVAC Chillers". In: *Automation Science and Engineering, IEEE Transactions on* 4.3 (2007), pp. 469–473.

[2] S. Zhang, K. Pattipati, Z. Hu, X. Wen, and C. Sankavaram. "Dynamic Coupled Fault Diagnosis With Propagation and Observation Delays". In: *Systems, Man, and Cybernetics: Systems, IEEE Transactions on* 43.6 (2013), pp. 1424–1439.

[3] S. Tornil-Sin, C. Ocampo-Martinez, V. Puig, and T. Escobet. "Robust Fault Diagnosis of Nonlinear Systems Using Interval Constraint Satisfaction and Analytical Redundancy Relations". In: *Systems, Man, and Cybernetics: Systems, IEEE Transactions on* 44.1 (2014), pp. 18–29.

[4] A. Scacchioli, G. Rizzoni, M. Salman, W. Li, S. Onori, and X. Zhang. "Model-based Diagnosis of an Automotive Electric Power Generation and Storage System". In: *Systems, Man, and Cybernetics: Systems, IEEE Transactions on* 44.1 (2014), pp. 72–85.

[5] R. Kwong and D. Yonge-Mallo. "Fault Diagnosis in Discrete-Event Systems with Incomplete Models: Learnability and Diagnosability". In: *Cybernetics, IEEE Transactions on* 45.7 (2015), pp. 1236–1249.

[6] S. Genc and S. Lafortune. "Distributed Diagnosis of Place-Bordered Petri Nets". In: *Automation Science and Engineering, IEEE Transactions on* 4.2 (2007), pp. 206–219.

[7] W. Qiu, Q. Wen, and R. Kumar. “Decentralized Diagnosis of Event-Driven Systems for Safely Reacting to Failures”. In: *Automation Science and Engineering, IEEE Transactions on* 6.2 (2009), pp. 362–366.

[8] R. Kumar and S. Takai. “Inference-Based Ambiguity Management in Decentralized Decision-Making: Decentralized Diagnosis of Discrete-Event Systems”. In: *Automation Science and Engineering, IEEE Transactions on* 6.3 (2009), pp. 479–491.

[9] Q. Liu, S. Qin, and T. Chai. “Decentralized Fault Diagnosis of Continuous Annealing Processes Based on Multilevel PCA”. In: *Automation Science and Engineering, IEEE Transactions on* 10.3 (2013), pp. 687–698.

[10] D. Lefebvre. “Fault Diagnosis and Prognosis With Partially Observed Petri Nets”. In: *Systems, Man, and Cybernetics: Systems, IEEE Transactions on* 44.10 (2014), pp. 1413–1424.

[11] M. Schwabacher and K. Goebel. “A survey of artificial intelligence for prognostics”. In: *AAAI Fall Symposium*. LA USA, 2007, pp. 107–114.

[12] B. Zhang, C. Sconyers, C. Byington, R. Patrick, M. Orchard, and G. Vachtsevanos. “A Probabilistic Fault Detection Approach: Application to Bearing Fault Detection”. In: *Industrial Electronics, IEEE Transactions on* 58.5 (2011), pp. 2011–2018.

[13] C. Chen, D. Brown, C. Sconyers, B. Zhang, G. Vachtsevanos, and M. E. Orchard. “An integrated architecture for fault diagnosis and failure prognosis of complex engineering systems”. In: *Expert Systems with Applications* 39.10 (2012), pp. 9031 –9040.

[14] Y. Ren, A. Wang, and H. Wang. “Fault Diagnosis and Tolerant Control for Discrete Stochastic Distribution Collaborative Control Systems”. In:

Systems, Man, and Cybernetics: Systems, IEEE Transactions on 45.3 (2015), pp. 462–471.

[15] B. Olivares, M. Cerda Munoz, M. Orchard, and J. Silva. "Particle-Filtering-Based Prognosis Framework for Energy Storage Devices With a Statistical Characterization of State-of-Health Regeneration Phenomena". In: *Instrumentation and Measurement, IEEE Transactions on* 62.2 (2013), pp. 364–376.

[16] D. Pola, H. Navarrete, M. Orchard, R. Rabie, M. Cerda, B. Olivares, J. Silva, P. Espinoza, and A. Perez. "Particle-Filtering-Based Discharge Time Prognosis for Lithium-Ion Batteries With a Statistical Characterization of Use Profiles". In: *Reliability, IEEE Transactions on* 64.2 (2015), pp. 710–720.

[17] W. Xian, B. Long, M. Li, and H. Wang. "Prognostics of Lithium-Ion Batteries Based on the Verhulst Model, Particle Swarm Optimization and Particle Filter". In: *Instrumentation and Measurement, IEEE Transactions on* 63.1 (2014), pp. 2–17.

[18] B. Zhang, L. Tang, J. DeCastro, M. Roemer, and K. Goebel. "Autonomous vehicle battery state-of-charge prnostics enhanced misison planning". In: *International Jouranl of Prognostics and Health Management* 5.2 (2014), pp. 1–12.

[19] B. Saha, K. Goebel, S. Poll, and J. Christophersen. "Prognostics methods for battery health monitoring using a Bayesian framework". In: *IEEE Transactions on Instrumentation and Measurement* 58.2 (2009), pp. 291–296.

[20] B. Pattipati, K. Pattipati, J. P. Christopherson, S. M. Namburu, D. V. Prokhorov, and L. Qiao. *Automotive battery management systems*. IEEE, 2008.

[21] Y.-H. Chiang and W.-Y. Sean. "Dynamical estimation of state-of-health of batteries by using adaptive observer". In: *Power Electronics and Intelligent*

Transportation System (PEITS), 2009 2nd International Conference on. Vol. 1. IEEE. 2009, pp. 110–115.

[22] D. Haifeng, W. Xuezhe, and S. Zechang. "A new SOH prediction concept for the power lithium-ion battery used on HEVs". In: *2009 IEEE Vehicle Power and Propulsion Conference*. IEEE. 2009, pp. 1649–1653.

[23] K. W. E. Cheng, B. Divakar, H. Wu, K. Ding, and H. F. Ho. "Battery-management system (BMS) and SOC development for electrical vehicles". In: *IEEE Transactions on Vehicular Technology* 60.1 (2011), pp. 76–88.

[24] R. Spotnitz. "Simulation of capacity fade in lithium-ion batteries". In: *Journal of Power Sources* 113.1 (2003), pp. 72–80.

[25] B. Saha and K. Goebel. "Modeling Li-ion battery capacity depletion in a particle filtering framework". In: *Proceedings of the annual conference of the prognostics and health management society*. 2009, pp. 2909–2924.

[26] B. Liuaw, R. Jungst, G Nagasubramanian, H. Case, and D. Doughty. "Modeling capacity fade in lithium-ion batteries". In: *J. Power Sources* 140 (2005), p. 157.

[27] U. Tröltzsch, O. Kanoun, and H.-R. Tränkler. "Characterizing aging effects of lithium ion batteries by impedance spectroscopy". In: *Electrochimica Acta* 51.8 (2006), pp. 1664–1672.

[28] N. D. Williard. "Degradation Analysis and Health Monitering of Lithium Ion Batteries". PhD thesis. 2011.

[29] I. Tumer and A. Bajwa. "A survey of aircraft engine health monitoring systems". In: *Proceedings of the 35th AIAA/ASME/ SAE/ASEE Joint Propulsion Conference*. 2004, pp. 620–625.

[30] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley and Sons, 2006.

[31] S. Yin, S. Ding, X. Xie, and H. Luo. "A review on basic data-driven approaches for industrial process monitoring". In: *IEEE Transactions on Industrial Electronics* 61.11 (2014), pp. 6418 –6428.

[32] S. Yin, X. Zhu, and O. Kaynak. "Improved PLS Focused on Key-Performance Indicator-Related Fault Diagnosis". In: *IEEE Transactions on Industrial Electronics* 62.3 (2015), pp. 1651–1658.

[33] A. Hess, G. Calvello, and P. Frith. "Challenges, issues, and lessons learned chasing the "Big P". Real predictive prognostics. Part 1". In: *2005 IEEE Aerospace Conference*. 2005, pp. 3610–3619.

[34] G. Vachtsevanos, L. Frank, R. Michael, H. Andrew, and W. Biqing. *Intelligent fault diagnosis and prognosis for engineering systems*. 2006.

[35] L. Jaw and W. Wang. "A run-time test system for maturing intelligent system/vehicle capabilities-SIDAL". In: *Aerospace Conference, 2004. Proceedings. 2004 IEEE*. Vol. 6. IEEE. 2004, pp. 3756–3763.

[36] A. K. Jardine, D. Lin, and D. Banjevic. "A review on machinery diagnostics and prognostics implementing condition-based maintenance". In: *Mechanical systems and signal processing* 20.7 (2006), pp. 1483–1510.

[37] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou. "Remaining useful life estimation–A review on the statistical data driven approaches". In: *European Journal of Operational Research* 213.1 (2011), pp. 1–14.

[38] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel. "Prognostics and health management design for rotary machinery systems-Reviews, methodology and applications". In: *Mechanical systems and signal processing* 42.1 (2014), pp. 314–334.

[39] R. Isermann. “Model-based fault-detection and diagnosis–status and applications”. In: *Annual Reviews in control* 29.1 (2005), pp. 71–85.

[40] S. Ding. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media, 2008.

[41] M Luo, D Wang, M Pham, C. Low, J. Zhang, D. Zhang, and Y. Zhao. “Model-based fault diagnosis/prognosis for wheeled mobile robots: a review”. In: *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005*. IEEE. 2005, 6–pp.

[42] L. Serrao, S. Onori, G. Rizzoni, and Y. Guezennec. “A novel model-based algorithm for battery prognosis”. In: *IFAC Proceedings Volumes* 42.8 (2009), pp. 923–928.

[43] V. H. Johnson, A. A. Pesaran, and T. Sack. *Temperature-dependent battery models for high-power lithium-ion batteries*. Citeseer, 2001.

[44] M. Chen and G. A. Rincon-Mora. “Accurate electrical battery model capable of predicting runtime and IV performance”. In: *IEEE transactions on energy conversion* 21.2 (2006), pp. 504–511.

[45] P. Ramadass, B. Haran, P. M. Gomadam, R. White, and B. N. Popov. “Development of First Principles Capacity Fade Model for Li-Ion Cells”. In: *Journal of The Electrochemical Society* 151.2 (2004), A196–A203.

[46] B. S. Haran, B. N. Popov, and R. E. White. “Determination of the hydrogen diffusion coefficient in metal hydrides by impedance spectroscopy”. In: *Journal of Power Sources* 75.1 (1998), pp. 56–63.

[47] G. Ning and B. N. Popov. “Cycle Life Modeling of Lithium-Ion Batteries”. In: *Journal of The Electrochemical Society* 151.10 (2004), A1584–A1591.

[48] C. Y. Wang, W. B. Gu, and B. Y. Liaw. "MicroâĂŘMacroscopic Coupled Modeling of Batteries and Fuel Cells: I. Model Development". In: *Journal of The Electrochemical Society* 145.10 (1998), pp. 3407–3417.

[49] V. R. Subramanian, D. Tapriyal, and R. E. White. "A boundary condition for porous electrodes". In: *Electrochemical and solid-state letters* 7.9 (2004), A259–A263.

[50] J. D. Kozlowski. "Electrochemical cell prognostics using online impedance measurements and model-based data fusion techniques". In: *Aerospace Conference, 2003. Proceedings. 2003 IEEE*. Vol. 7. IEEE. 2003, pp. 3257–3270.

[51] B. Saha, K. Goebel, and J. Christophersen. "Comparison of prognostic algorithms for estimating remaining useful life of batteries". In: *Transactions of the Institute of Measurement and Control* (2009).

[52] D. Liu, Y. Luo, Y. Peng, X. Peng, and M. Pecht. "Lithium-ion battery remaining useful life estimation based on nonlinear AR model combined with degradation feature". In: *Annual Conference of the Prognostics and Health Management Society*. Vol. 3. 2012, pp. 1803–1836.

[53] D. Liu, Y. Luo, J. Liu, Y. Peng, L. Guo, and M. Pecht. "Lithium-ion battery remaining useful life estimation based on fusion nonlinear degradation AR model and RPF algorithm". In: *Neural Computing and Applications* 25.3-4 (2014), pp. 557–572.

[54] A. J. Salkind, C. Fennie, P. Singh, T. Atwater, and D. E. Reisner. "Determination of state-of-charge and state-of-health of batteries by fuzzy logic methodology". In: *Journal of Power Sources* 80.1 (1999), pp. 293–300.

[55] J. Liu, A. Saxena, K. Goebel, B. Saha, and W. Wang. *An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries*. Tech. rep. DTIC Document, 2010.

[56] L. Peel. “Data driven prognostics using a Kalman filter ensemble of neural network models”. In: *Prognostics and Health Management, 2008. PHM 2008. International Conference on*. IEEE. 2008, pp. 1–6.

[57] R. Huang, L. Xi, X. Li, C. R. Liu, H. Qiu, and J. Lee. “Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods”. In: *Mechanical Systems and Signal Processing* 21.1 (2007), pp. 193–207.

[58] B. Pattipati, C. Sankavaram, and K. Pattipati. “System identification and estimation framework for pivotal automotive battery management system characteristics”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 41.6 (2011), pp. 869–884.

[59] B. Saha, S. Poll, K. Goebel, and J. Christophersen. “An integrated approach to battery health monitoring using Bayesian regression and state estimation”. In: *2007 IEEE Autotestcon*. IEEE. 2007, pp. 646–653.

[60] J. V. Tu. “Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes”. In: *Journal of clinical epidemiology* 49.11 (1996), pp. 1225–1231.

[61] P. A. Cerny and M. A. Proximity. “Data mining and neural networks from a commercial perspective”. In: *ORSNZ Conference Twenty Naught One*. 2001.

[62] S. Butler. “Prognostic Algorithms for Condition Monitoring and Remaining Useful Life Estimation”. PhD thesis. NATIONAL UNIVERSITY OF IRELAND, MAYNOOTH, 2012.

[63] M. E. Tipping. “Sparse Bayesian learning and the relevance vector machine”. In: *Journal of machine learning research* 1.Jun (2001), pp. 211–244.

[64] D. Wang, Q. Miao, and M. Pecht. “Prognostics of lithium-ion batteries based on relevance vectors and a conditional three-parameter capacity degradation model”. In: *Journal of Power Sources* 239 (2013), pp. 253–264.

[65] J. Quinonero-Candela and L. K. Hansen. “Time series prediction based on the relevance vector machine with adaptive kernels”. In: *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*. Vol. 1. IEEE. 2002, pp. I–985.

[66] M. E. Orchard and G. J. Vachtsevanos. “A particle filtering approach for on-line failure prognosis in a planetary carrier plate”. In: *International Journal of Fuzzy Logic and Intelligent Systems* 7.4 (2007), pp. 221–227.

[67] M. E. Orchard and G. J. Vachtsevanos. “A particle-filtering approach for on-line fault diagnosis and failure prognosis”. In: *Transactions of the Institute of Measurement and Control* (2009).

[68] Q. Miao, L. Xie, H. Cui, W. Liang, and M. Pecht. “Remaining useful life prediction of lithium-ion battery with unscented particle filter technique”. In: *Microelectronics Reliability* 53.6 (2013), pp. 805–810.

[69] S. J. Julier and J. K. Uhlmann. “New extension of the Kalman filter to nonlinear systems”. In: *AeroSense’97*. International Society for Optics and Photonics. 1997, pp. 182–193.

[70] E. A. Wan and R. Van Der Merwe. “The unscented Kalman filter for nonlinear estimation”. In: *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. Ieee. 2000, pp. 153–158.

[71] F. Zhang, G. Liu, and L. Fang. “Battery state estimation using unscented kalman filter”. In: *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE. 2009, pp. 1863–1868.

[72] S. Santhanagopalan and R. E. White. “State of charge estimation using an unscented filter for high power lithium ion cells”. In: *International Journal of Energy Research* 34.2 (2010), pp. 152–163.

[73] G. L. Plett. “Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification”. In: *Journal of power sources* 134.2 (2004), pp. 262–276.

[74] G. L. Plett. “Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation”. In: *Journal of Power sources* 134.2 (2004), pp. 277–292.

[75] B. Bhangu, P. Bentley, D. A. Stone, and C. M. Bingham. “Nonlinear observers for predicting state-of-charge and state-of-health of lead-acid batteries for hybrid-electric vehicles”. In: *IEEE Transactions on Vehicular Technology* 54.3 (2005), pp. 783–794.

[76] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking”. In: *IEEE Transactions on signal processing* 50.2 (2002), pp. 174–188.

[77] A. Doucet and A. M. Johansen. “A tutorial on particle filtering and smoothing: Fifteen years later”. In: *Handbook of nonlinear filtering* 12.656-704 (2009), p. 3.

[78] D. Di Domenico, G. Fiengo, and A. Stefanopoulou. “Lithium-ion battery state of charge estimation with a Kalman filter based on a electrochemical model”. In: *2008 IEEE International Conference on Control Applications*. Ieee. 2008, pp. 702–707.

[79] A. Vasebi, M. Partovibakhsh, and S. M. T. Bathaee. “A novel combined battery model for state-of-charge estimation in lead-acid batteries based on extended Kalman filter for hybrid electric vehicle applications”. In: *Journal of Power Sources* 174.1 (2007), pp. 30–40.

[80] M. Gao, Y. Liu, and Z. He. "Battery state of charge online estimation based on particle filter". In: *Image and Signal Processing (CISP), 2011 4th International Congress on*. Vol. 4. IEEE. 2011, pp. 2233–2236.

[81] M. E. Orchard, P. Hevia-Koch, B. Zhang, and L. Tang. "Risk measures for particle-filtering-based state-of-charge prognosis in lithium-ion batteries". In: *IEEE Transactions on Industrial Electronics* 60.11 (2013), pp. 5260–5269.

[82] J. Taylor. *Introduction to error analysis, the study of uncertainties in physical measurements*. Vol. 1. 1997.

[83] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel. "Metrics for offline evaluation of prognostic performance". In: *International Journal of Prognostics and Health Management* 1.1 (2010), p. 20.

[84] W. He, N. Williard, M. Osterman, and M. Pecht. "Prognostics of Lithium-ion batteries based on Dempster-Shafer theory and the Bayesian Monte Carlo method". In: *Journal of Power Sources* 196 (2011), pp. 1034–10321.

[85] R. McCann and A. Le. "Lebesgue Sampling with a Kalman Filter in Wireless Sensors for Smart Appliance Networks". In: *Proceedings of Industry Applciaitons Society Annual Meeting*. 2008.

[86] K. Astrom and B. Bernhardsson. "Comparison of Riemann and Lebesgue sampling for first order stochastic systems". In: *Proceedings of IEEE Conference on Decision and Control*. 1999.

[87] R. Morales-Menendez, N. de Freitas, I. Monterrey, O. D. Freitas, and D. Poole. "Real-Time Monitoring of Complex Industrial Processes With Particle Filters". In: *NIPS*. 2002, pp. 1433–1440.

[88] N. de Freitas. "Rao-Blackwellised particle filtering for fault diagnosis". In: *Proceedings of the IEEE Aerospace Conference*. Vol. 4. 2002, pp. 1767–1772.

[89] B. Zhang, T. Khawaja, R. Patrick, and G. Vachtsevanos. "A novel blind deconvolution de-noise scheme in failure prognosis". In: *Transactions of the Institute of Measurement and Control* 32.1 (2010), pp. 3–30.

[90] W. He, N. Williard, M. Osterman, and M. Pecht. "Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method". In: *Journal of Power Sources* 196.23 (2011), pp. 10314–10321.

[91] B. Zhang and X. Wang. "Fault Diagnosis and Prognosis Based on Lebesgue Sampling". In: *Annual Conference of the Prognostics and Health Management Society 2014*. Vol. 5. 2014.

[92] W. Yan, B. Zhang, X. Wang, W. Dou, and J. Wang. "Lebesgue-Sampling-Based Diagnosis and Prognosis for Lithium-Ion Batteries". In: *IEEE Transactions on Industrial Electronics* 63.3 (2016), pp. 1804–1812.

[93] M. Orchard, G. Kacprzynski, K. Goebel, B. Saha, and G. Vachtsevanos. "Advances in uncertainty representation and management for particle filtering applied to prognostics". In: *International Conference on Prognostics and Health Management*. Denver CO, 2008.

[94] J. Celaya, A. Saxena, and K. Goebel. "Uncertainty representation and interpretation in model-based prognostics algorithms based on Kalman filter estimation". In: *Proceedings of the Annual Conference of the Prognostics and Health Management Society*. Minneapolis, MN, 2012.

[95] S. Sankararaman, M. Daigle, and K. Goebel. "Uncertainty Quantification in Remaining Useful Life Prediction Using First-Order Reliability Methods". In: *Reliability, IEEE Transactions on* 63.2 (2014), pp. 603–619.

[96] B. Saha and K. Goebel. “Uncertainty management for diagnostics and prognostics of batteries using Bayesian techniques”. In: *Proceedings of IEEE Aerospace Conference*. 2008.

[97] Y. Li, J. Chen, and L. Feng. “Dealing with Uncertainty: A Survey of Theories and Practices”. In: *Knowledge and Data Engineering, IEEE Transactions on* 25.11 (2013), pp. 2463–2482.

[98] X. Wang and B. Zhang. “Real-time lebesgue-sampled model for continuous-time nonlinear systems”. In: *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. 2014, pp. 4367–4372.

[99] A. Sidhu, A. Izadian, and S. Anwar. “Adaptive nonlinear model-based fault diagnosis of Li-ion batteries”. In: *IEEE Transactions on Industrial Electronics* 62.2 (2015), pp. 1002–1011.

[100] S. Piller, M. Perrin, and A. Jossen. “Methods for state-of-charge determination and their applications”. In: *Journal of power sources* 96.1 (2001), pp. 113–120.

[101] In: URL: `http://batteryuniversity.com/learn/article/how_to_monitor_a_battery`.

[102] H. Dai, Z. Sun, and X. Wei. “Online SOC estimation of high-power lithium-ion batteries used on HEVs”. In: *Vehicular Electronics and Safety, 2006. ICVES 2006. IEEE International Conference on*. IEEE. 2006, pp. 342–347.

[103] Z. He, M. Gao, and J. Xu. “EKF-Ah based state of charge online estimation for lithium-ion power battery”. In: *Computational Intelligence and Security, 2009. CIS’09. International Conference on*. Vol. 1. IEEE. 2009, pp. 142–145.

[104] J. Xu, M. Gao, Z. He, Q. Han, and X. Wang. “State of charge estimation online based on EKF-Ah method for lithium-ion power battery”. In: *Image and Signal Processing, 2009. CISP’09. 2nd International Congress on*. IEEE. 2009, pp. 1–5.

[105] Y. He, X. Liu, C. Zhang, and Z. Chen. “A new model for State-of-Charge (SOC) estimation for high-power Li-ion batteries”. In: *Applied Energy* 101 (2013), pp. 808–814.

[106] H. Dai, X. Wei, Z. Sun, J. Wang, and W. Gu. “Online cell {SOC} estimation of Li-ion battery packs using a dual time-scale Kalman filtering for {EV} applications”. In: *Applied Energy* 95 (2012), pp. 227–237.

[107] O. Barbarisi, F. Vasca, and L. Glielmo. “State of charge Kalman filter estimator for automotive batteries”. In: *Control Engineering Practice* 14.3 (2006), pp. 267–275.

[108] M. Charkhgard and M. Farrokhi. “State-of-charge estimation for lithium-ion batteries using neural networks and EKF”. In: *IEEE transactions on industrial electronics* 57.12 (2010), pp. 4178–4187.

[109] W.-Y. Chang. “The state of charge estimating methods for battery: A review”. In: *ISRN Applied Mathematics* 2013 (2013).

[110] P. Lall, R. Lowe, and K. Goebel. “Extended Kalman Filter models and resistance spectroscopy for prognostication and health monitoring of leadfree electronics under vibration”. In: *Prognostics and Health Management (PHM), 2011 IEEE Conference on*. 2011, pp. 1–12.

[111] P. Lall, J. Wei, and K. Goebel. “Comparison of Lalman-filter and extended Kalman-filter for prognostics health management of electronics”. In: *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2012 13th IEEE Intersociety Conference on*. IEEE. 2012, pp. 1281–1291.

[112] S. Pang, J. Farrell, J. Du, and M. Barth. “Battery state-of-charge estimation”. In: *American Control Conference, 2001. Proceedings of the 2001*. Vol. 2. IEEE. 2001, pp. 1644–1649.

[113] J. Chiasson and B. Vairamohan. “Estimating the state of charge of a battery”. In: *American Control Conference, 2003. Proceedings of the 2003*. Vol. 4. 2003, pp. 2863–2868.

[114] M. Coleman, C. K. Lee, C. Zhu, and W. G. Hurley. “State-of-charge determination from EMF voltage estimation: Using impedance, terminal voltage, and current for lead-acid and lithium-ion batteries”. In: *IEEE Transactions on industrial electronics* 54.5 (2007), pp. 2550–2557.

[115] G Plett. “Dual and joint EKF for simultaneous SOC and SOH estimation”. In: *Proceedings of the 21st Electric Vehicle Symposium (EVS21), Monaco*. 2005, pp. 1–12.

[116] P. Moss, G Au, E. Plichta, and J. Zheng. “An electrical circuit for modeling the dynamic response of li-ion polymer batteries”. In: *Journal of The Electrochemical Society* 155.12 (2008), A986–A994.

[117] X. Tang, X. Mao, J. Lin, and B. Koch. “Li-ion battery parameter estimation for state of charge”. In: *American Control Conference (ACC), 2011*. IEEE. 2011, pp. 941–946.

[118] K Amine, C. Chen, J Liu, M Hammond, A Jansen, D Dees, I Bloom, D Vissers, and G Henriksen. “Factors responsible for impedance rise in high power lithium ion batteries”. In: *Journal of Power Sources* 97 (2001), pp. 684–687.